

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL  
INSTITUTO POLITÉCNICO NACIONAL

DEPARTAMENTO DE CONTROL AUTOMÁTICO

## **Redes neuronales difusas dinámicas para identificación y control adaptable**

TESIS QUE PRESENTA  
**Rigoberto Toxqui Toxqui**

PARA OBTENER EL GRADO DE  
**MAESTRO EN CIENCIAS**

EN LA ESPECIALIDAD DE  
**CONTROL AUTOMÁTICO**

DIRECTORES DE TESIS:  
**Dr. Wen Yu Liu**

**México, D.F., abril del 2003.**



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	3
1.2. Objetivos . . . . .	3
1.3. Publicaciones . . . . .	4
1.4. Estructura de la tesis . . . . .	4
<b>2. Neuro difuso identificación y control</b>	<b>7</b>
2.1. Redes neuronales . . . . .	7
2.1.1. La neurona . . . . .	9
2.1.2. Topología de la red neuronal (RN) . . . . .	12
2.2. Lógica difusa . . . . .	13
2.2.1. Conjuntos difusos . . . . .	14
2.2.2. Borrosificador (Fuzzifier) . . . . .	16
2.2.3. Desborrosificador (Defuzzifier) . . . . .	17
2.3. Neuro difuso (Neurofuzzy) . . . . .	19
2.3.1. Estática y dinámica en sistemas difusos . . . . .	20
2.3.2. Comparaciones generales de sistemas difuso y redes neuronales . . . . .	21
2.4. La estabilidad de la entrada a estado (ISS) y funciones de Lyapunov ISS . . . . .	22
<b>3. Estática neuro difuso para identificación</b>	<b>25</b>
3.1. Estática difusa redes neuronales . . . . .	25

3.1.1.	Estructura tipos mamdani estática difusa redes neuronales . . . . .	29
3.1.2.	Estructura tipos takagi sugeno kang estática difusa redes neuronales .	31
3.2.	Sistemas de identificación difusa . . . . .	32
3.3.	Sistemas de identificación difusa neuronal . . . . .	35
3.3.1.	Funciones pertenencia conocidas . . . . .	35
3.3.2.	Funciones pertenencia desconocidas . . . . .	37
3.4.	Simulación . . . . .	38
<b>4.</b>	<b>Dinámica neuro difuso para identificación</b>	<b>49</b>
4.1.	Dinámica difusa redes neuronales . . . . .	49
4.1.1.	Estructura tipo mamdani dinámica difusa redes neuronales . . . . .	49
4.1.2.	Estructura tipo takagi sugeno dinámica difusa redes neuronales . . .	52
4.2.	Sistemas de identificación con dinámica neuro difuso . . . . .	53
4.2.1.	Funciones pertenencia conocidas . . . . .	53
4.2.2.	Funciones pertenencia desconocidas . . . . .	55
4.3.	Simulación . . . . .	61
<b>5.</b>	<b>Redes neuronales difusas dinámicas para sistemas no lineales de control adaptable</b>	<b>67</b>
5.1.	Estructura difusa neuro control . . . . .	67
5.2.	Difusa neuro control con algoritmo estable . . . . .	71
5.2.1.	Aprendizaje sin funciones pertenencia . . . . .	71
5.2.2.	Aprendizaje con funciones pertenencia . . . . .	73
5.3.	Simulación . . . . .	77
<b>6.</b>	<b>Conclusiones</b>	<b>83</b>

# Índice de figuras

2.1. Modelo de una neurona . . . . .	10
2.2. a) Escalón, b) Saturación c) Sigmoide d) Tangente hiperbólica . . . . .	13
2.3. Arquitectura básica de un controlador de lógica difusa (FLC) . . . . .	16
2.4. Estructura se un sistema de lógica difusa . . . . .	18
2.5. Representación de la red de un sistema difuso con una red neuronal de tres capas . . . . .	19
3.1. Arquitectura de un sistema neuronal difuso . . . . .	31
3.2. Selección de variables de proposición . . . . .	35
3.3. Sistema . . . . .	40
3.4. Datos de consecuencia y ruido . . . . .	41
3.5. Función gaussiana . . . . .	41
3.6. Aproximación de la función . . . . .	42
3.7. Aprendizaje inestable . . . . .	43
3.8. Error medio cuadratico . . . . .	44
3.9. Modelado de un sistema no lineal . . . . .	46
3.10. Error cuadratico medio . . . . .	47
4.1. La estructura del sistema de la dinámica neuronal difuso . . . . .	51
4.2. Sistema dinámico caótico de Lorenz . . . . .	64
4.3. Sistema dinámico caótico de Lorenz . . . . .	65
4.4. Sistema dinámico caótico de Lorenz . . . . .	66

5.1. Identificación del sistema no lineal para el modelo Chen . . . . .	79
5.2. Identificación del sistema no lineal para el modelo Narendra . . . . .	80
5.3. Identificación del sistema no lineal para el modelo Wen . . . . .	81

# Capítulo 1

## Introducción

Las redes neuronales han sido populares modelos en una extensa variedad en modelado y control, sin embargo las redes neuronales no pueden incluir descripciones en sistemas lingüísticos, por que estos métodos se dificultan para obtener el conocimiento en una manera comprensiva. Por otro lado, la deducción de los sistemas difusos representan el conocimiento usando etiquetas lingüísticas y quizás sea más fácil de interpretar. Pero los sistemas difusos no tienen mucha capacidad de aprendizaje, es difícil para un operador humano poner a punto las reglas difusas y las funciones de pertenencia de un conjunto de datos de entrenamiento. Entonces prometiéndolo aproximar para combinar dentro de un sistema integrado, tal que podemos obtener resultados de ambos sistemas difusos y redes neuronales.

La aproximación del sistema neuro difuso tiene puestos muchas atenciones recientemente. La fusión de las dos diferentes tecnologías pueden realizarse en tres direcciones [14].

1. Los sistemas neuronales difusos (redes neuronales basados en sistemas difusos), usan redes neuronales como herramienta para los modelos difusos.
2. Las redes neuronales difusas (modelos de redes neuronales basados en lógica difusa), se utilizan para borrosificadores de los modelos de redes neuronales convencionales.
3. Los sistemas híbridos neuronales difusos, unión de tecnología de lógica difusa y redes neuronales dentro de sistemas híbridos.

Los primeros dos sistemas representan a favor combinaciones, donde una tecnología esta presente en la otra. La tercera muestra contribuciones de combinaciones, donde las dos tecnologías son incluidos para realizar una común tarea.

Estos modelos libres se aproximan usando una característica de las redes neuronales y sistemas difusos, pero la falta del modelo hace difícil obtener resultados teóricos en la estabilidad de sistemas neuro difusos. Para el diseño es muy importante asegurar la teoría de estabilidad antes de que estos sistemas quieran aplicarse a controladores neuro difuso a sistemas reales. A pesar de sus aplicaciones exitosas, no hay muchos resultados en el análisis de estabilidad para sistemas neuro difusos ahí aparecen un número de análisis de estabilidad y síntesis de resultados en control difuso. La idea básica de estos métodos es diseñar un controlador de retroalimentación o retropropagación (BP) de cada modelo local y la construcción de un controlador global del controlador local de tal manera que la estabilidad global de un sistema de control difuso en lazo cerrado sea garantizado. La existencia de un controlador difuso para el sistema no lineal puede ser estabilizado con la teoría de lógica difusa.

Las redes neuronales recurrentes han sido estudiados durante las últimas décadas. El análisis de Lyapunov es apropiado para redes neuronales dinámicas, las redes neuronales de capa simple fueron discutidos en [36], [39]. Las redes de alto orden y las redes multicapas se encuentran en [37], [38]. La estabilidad de algoritmos de aprendizaje pueden ser derivados por el análisis de la identificación o error de seguimiento de las redes neuronales.

Desde que las redes neuronales no pueden igualar a los sistemas desconocidos no lineales exactamente, algunas modificaciones robustas [21] deberían ser aplicados en el gradiente normal o en los algoritmos de retroalimentación [36], [39]. La estabilidad de entrada a estado (ISS) y la aproximación de pasividad son herramientas alternativas efectivas para redes neuronales dinámicas. Las propiedades de pasividad de redes neuronales dinámicas se describen en [40].

## 1.1. Motivación

Básicamente la tarea a realizar es un modelo neuronal difuso para sistemas dinámicos de identificación y control utilizando un algoritmo de aprendizaje. En artículos publicados por [6], se analiza el modelado difuso para identificación de sistemas dinámicos en particular se describe un nuevo modelo difuso de una red neuronal difusa dinámica (DFNN), consta de reglas recurrentes Takagi Sugeno Kang (TSK). El objetivo principal parte del borrosificador estático, mientras las siguientes partes de la reglas difusas son redes neuronales recurrentes con la retroalimentación interior y la sinapsis de retraso de tiempo. La red está entrenada por medio de un nuevo algoritmo de aprendizaje, el método de optimización restringiendo la dinámica neuronal difuso (D-FUNCOM), basado en el concepto de restringir la optimización. La propuesta del modelo dinámico, provee con el algoritmo de aprendizaje, se aplican a varios problemas temporales, mientras se incluyen procesos de modelado tipo NARMA y el problema de la cancelación de ruido. Comparando con la función, estas llevan a series estáticos, sistemas dinámicos y algunos modelos difusos existentes recurrentes.

En general las redes neuronales difusas no pueden igualar a los sistemas no lineales. La dinámica del no modelado lleva la tendencia de los parámetros al igual que el problema de estabilidad. Algunas modificaciones robustas han sido incluidos en orden para garantizar la estabilidad de Lyapunov. En esta tesis se aproxima la estabilidad entrada a estado, es aplicado a acceso de algoritmos de entrenamiento robusto en redes neuronales difusas. Se propone el análisis del modelado neuronal difuso en esta tesis no necesitan una modificación robusta y esta robustes en alguna inestabilidad acotada.

## 1.2. Objetivos

1. El objetivo de esta tesis es abordar el modelo neuronal difuso para identificación y control, para sistemas dinámicos y estáticos con funciones de pertenencia conocidas y funciones de pertenencia desconocidas, así como el análisis de redes neuronales recurrentes difusas para sistemas no lineales de control.

- Modelo neuronal difuso recurrente de tipo takagi sugeno kang (TSK).
  - Modelo neuronal difuso recurrente de tipo mandani.
2. Algoritmos de aprendizaje estables para dos modelos.
  3. Control estable adaptable usando sistemas recurrentes neuro difuso.

### 1.3. Publicaciones

Parte de esta tesis se presentó como el artículo titulado "Stable Fuzzy Identification Using Recurrent Fuzzy Neural Networks" en el *IASTED International Conference Neural Networks and Computational Intelligence*, 2003, Cancun, México

### 1.4. Estructura de la tesis

El presente trabajo consta de seis capítulos donde se expone la importancia del trabajo.

En el capítulo dos se presenta la estrategia de identificación para un sistema neuro difuso de identificación y control, definiendo redes neuronales, lógica difusa finalmente neuro difuso.

En el capítulo tres se describe el sistema estática neuro difuso para la identificación, describiendo la estática en redes neuronales para una red neuronal de capa simple y multicapa así como la estática difusa en redes neuronales describiendo el sistema para la estructura de las redes neuronales del tipo mandani y takagi sugeno kang. El aprendizaje de funciones de pertenencia conocidas y funciones de pertenencia desconocidas, posteriormente se presenta la parte de simulación.

En el capítulo cuatro se define la dinámica de identificación neuro difuso, primero redes neuronales dinámicas para sistemas de capa simple y multicapa en redes neuronales recurrentes, como segundo punto se describe la dinámica difusa de redes neuronales, estructura tipo mandani, takagi sugeno para funciones pertenencia conocidas y funciones pertenencia desconocidas finalmente se presenta la simulación.

En el capítulo cinco se hace el análisis del sistema de redes neuronales difusas dinámicas para sistemas no lineales de control adaptable, analizando su estructura difusa neuro control, análisis de estabilidad difusa neuro control con y sin funciones pertenencia de aprendizaje, también se tiene la parte de la simulación en donde se comparan tres métodos diferentes Chen, Narendra y Wen.

Finalmente en el capítulo seis se presentan las conclusiones generales del trabajo, así mismo se presentan propuestas para futuros trabajos.



# Capítulo 2

## Neuro difuso identificación y control

### 2.1. Redes neuronales

La creciente demanda en tecnología de nuestra sociedad requiere nuevos enfoques para los problemas actuales de control. Las Redes Neuronales (RN) con su masivo paralelismo y su capacidad para aprender prometen mejores soluciones, al menos para algunos problemas. En este momento, la comunidad de control ha puesto su atención en las RN y se pregunta si pueden ser usadas para obtener mejores soluciones para viejos problemas o quizá soluciones a problemas que se han resistido a nuestros mejores esfuerzos.

El uso de las RN en control puede verse como un paso natural en la evolución de la teoría de control. Viendo hacia atrás la evolución del control se ha visto motivada por tres aspectos fundamentales: la necesidad de tratar con sistemas complejos; la necesidad de cumplir con la creciente demanda de nuevos modelos de control y la necesidad de obtener esos modelos con el menor conocimiento de la planta y el medio ambiente, es decir la necesidad de controlar bajo incertidumbre. La necesidad de controlar sistemas dinámicos complejos bajo incertidumbre ha conducido a la reevaluación de los métodos convencionales de control, y por lo tanto la necesidad de nuevos métodos es obvia. Esto también ha conducido a un concepto mas general de control, uno que incluye decisión, planeación y aprendizaje, los cuales son cualidades necesarias cuando se desea un sistema con autonomía. En vista a esto, no es de sorprender

que la comunidad de control este muy activa buscando nuevas ideas para resolver de forma efectiva los problemas de control de la sociedad moderna. La necesidad es la madre de la invención, y en control no ha sido la excepción, y por ejemplo en el siglo III se diseño un reloj de agua con un mecanismo de retroalimentación, el primer artefacto con retroalimentación en la historia. Así el uso de las RN en control es un paso bastante natural en su evolución. Las RN parecen ofrecer un mejor entendimiento y quizás la solución de algunos de los problemas de control mas difíciles. Es claro que las RN pueden ser aceptadas y usadas si ellas resuelven problemas que han sido previamente imposible o muy difícil de resolver. Las RN pueden ser rechazadas y ser solo producto de la moda, como lo han sido inventos tales como la máquina de escribir eléctrica entre otros, si no promueven su utilidad. El reto es entonces encontrar el mejor camino para utilizar completamente esta nueva poderosa herramienta en control.

Al referirnos a RN en control es importante dar las características y propiedades más significativas de las RN:

- Las RN poseen una gran habilidad para la aproximación de funciones.
- Los elementos básicos de procesamiento en una RN tienen una estructura simple. Esto en conjunción con una implementación en paralelo resulta en un procesamiento muy rápido.
- La capacidad de aprendizaje y de generalización de las RN las convierte en una herramienta poderosa principalmente en sistemas variantes además de su capacidad de adaptación en línea.
- Las RN pueden operar con datos de forma cuantitativa así como de forma cualitativa.
- Las RN pueden procesar muchas entradas y también muchas salidas, por lo tanto son altamente aplicables en sistemas multivariables.

Desde el punto de vista de control la habilidad de las RN para tratar con sistemas no lineales es un punto muy significativo. Respecto a este tema métodos para el diseño de

controles no lineales que incluyen métodos teórico-prácticos y optimización juegan un rol importante y una posibilidad es el uso de RN para obtener sistemas de control no lineales.

Otra área en control es la teoría de control adaptable. Esta área ha producido muchos resultados teóricos en los últimos 20 años, sin embargo es solo relativo a la teoría de sistemas lineales, por lo tanto cuando estamos tratando con sistemas no lineales variantes en el tiempo los cuales son bastante mas complejos las RN son una buena opción.

Autores como Widrow (et. al.) [7] definen una red neuronal como un sistema con entradas y salidas que está compuesta de muchos elementos de procesamiento similares. Cada elemento de procesamiento tiene un número de pesos sinápticos o simplemente pesos. Ajustando los pesos de un elemento se puede cambiar el comportamiento del mismo y, por lo tanto, puede también alterar el comportamiento de total de la red para alcanzar la relación de entrada salida deseada. Este último proceso es conocido como entrenamiento de la red.

Para nuestro propósito una Red Neuronal (RN) la definimos como un aproximador no lineal de funciones compuesto como dijimos anteriormente de neuronas, organizadas en una topología específica que utiliza una ley de aprendizaje para mejorar su desempeño.

### 2.1.1. La neurona

La neurona o elemento de procesamiento a su vez están constituidas de: vector de entrada, pesos, umbral, un operador matemático de suma y función de activación, Figura (2.1).

El primer modelo matemático de una neurona biológica fue dado por Mc Culloch y Pitt [8] y es el siguiente:

$$y = \varphi\left(\sum_{i=0}^n w_i u_i\right) \quad (2.1)$$

donde:  $u_i$ ,  $i = 1, 2, \dots, n$  son las entradas a la neurona,  $u_0$  es la entrada al umbral e  $y$  es la salida de la neurona, los pesos sinápticos  $w_i \in \mathfrak{R}$ ,  $i = 1, 2, \dots, n$  cambian su valor según se requiera,  $\varphi(\cdot)$  es la función de activación. Lapedes Farber mostró, una arquitectura que es básicamente la misma que la de Mc Culloch y Pitt, esta es:

$$y(k) = \varphi(v(k)); \quad v(k) = \sum_{i=0}^n w_i u_i(k) \quad (2.2)$$

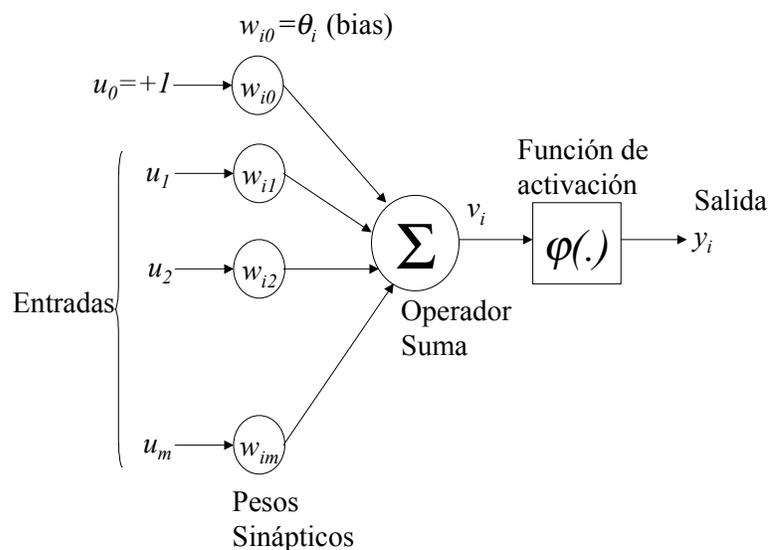


Figura 2.1: Modelo de una neurona

donde  $y$ ,  $\varphi$ ,  $w$  y  $u$  son definidas como en el párrafo anterior,  $k$  es la variable de tiempo. A continuación se da una descripción de cada uno de los elementos

### Vector de entrada

Es denotado por  $U$ ,  $U = (u_1, u_2, \dots, u_m)$  donde  $m$  es el número de entradas a la RN y a su vez la dimensión de  $U$ , este vector  $U$  son los datos con los que va operar la neurona, estas pueden ser dadas del medio ambiente o ser salidas de neuronas anteriores. Cabe hacer notar que si estamos hablando de identificación de funciones o control de sistemas  $U = U(k)$ , donde  $k \in K \subset \mathfrak{R}$ ; es la variable del tiempo, ya sea, según sea el caso, continuo o discreto, para esta situación las entradas generan una matriz  $E \in \mathfrak{R}^{n \times m}$  donde  $m$  es la dimensión del vector de entradas y  $n$  es la dimensión del subespacio  $K$ , esta  $n$  generalmente se le conoce como número de iteraciones.

### Pesos sinápticos

Al ser capturados los datos de entrada estos son propagados a través de la red, en el proceso de propagación cada componente  $u_i$  del vector de entrada  $U$  es multiplicada por una

variable  $w_{ij}$ , la cual aumenta o atenúa la señal de entrada, a  $w_{ij}$  se le conoce como peso sináptico o simplemente peso, estos pesos no tienen el mismo valor siempre sino que se van modificando según se requiera para tener un mejor desempeño, posteriormente puede haber una convergencia y entonces estar fijos. Cuando se habla de que una red es capaz de aprender se refiere al hecho de poder modificar sus pesos  $w_{ij}$ , el conjunto de pesos genera una matriz  $W$ , es decir,  $w_{ij}$  es la  $ij$ -ésima componente de la matriz de pesos  $W$ .

### El umbral

Denotado por  $\theta_i$  es simplemente un peso mas, pero a diferencia de estos el umbral tiene una entrada fija de (-1). En el contexto en que usaremos las RN, el umbral nos sirve para trasladar el punto de equilibrio de la RN, del origen  $\bar{x} = 0$  a un punto  $\bar{x} \neq 0$ .

### Operador de suma

Realiza la adición de los productos  $u_i * w_{ij}$ , la operación aquí descrita constituyen una combinación lineal generando un campo local inducido (CLI)  $v_i$ , es decir:

$$v_i = \sum_i u_i w_{ij} - \theta_i \quad (2.3)$$

### La función de activación

Denotada por  $\varphi(\cdot)$ , define la salida de la neurona en términos del CLI  $v_i$ , en [9] es definida como una función para limitar la amplitud de la salida neuronal. Sin embargo en el sentido en el que estamos tratando las Redes Neuronales, la razón de ser de la función de activación es la de proporcionar un comportamiento no lineal a la red y así poder aproximar funciones de todo tipo, el teorema universal de aproximación de funciones [9] da condiciones para determinar si una función  $\varphi$ , es una función de activación.

La función Sigmoide, Tangente Hiperbólica, Saturación o Escalón cumplen las condiciones del teorema y son utilizadas como funciones de activación. Las tres primeras funciones son aptas para nuestro propósito, el escalón como veremos mas adelante tiene una gran aplicación en reconocimiento de patrones.

**Función Escalón:** Para este tipo de función de activación tenemos que:

$$\varphi(v) = \begin{cases} 1 & \text{si } v \geq 0 \\ -1 & \text{si } v < 0 \end{cases} \quad (2.4)$$

donde  $v$  es el CLI.

**Función Saturación (Sat):** Esta definida por la siguiente ecuación:

$$\varphi(v) = \begin{cases} c & \text{si } v \geq +1 \\ v & \text{si } +1 > v > -1 \\ -c & \text{si } v \leq -1 \end{cases} \quad (2.5)$$

donde el factor de amplificación  $c$  es generalmente tomado como 1, pero su valor puede variar según sea requerido en las tareas de identificación y control.

**Función Sigmoide (Sigm):** Es de hecho la función más común en la construcción de RN. Esta es definida como una función estrictamente creciente que exhibe un balance entre una conducta lineal y no lineal y es definida por:

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (2.6)$$

donde  $a$  es un parámetro que define la pendiente de la función sigmoide.

**Función Tangente Hiperbólica (Tansig):** Al igual que la sigmoide es una de las principales funciones en la construcción de las RN especialmente como es mencionado en tareas de identificación y control:

$$\varphi(v) = \frac{1 - \exp(av)}{1 + \exp(-av)} \quad (2.7)$$

donde  $a$  es como antes, un parámetro. Las funciones escalón, saturación, sigmoide así como la tangente hiperbólica son mostradas en la Figura (2.2).

### 2.1.2. Topología de la red neuronal (RN)

Es la forma en que se organizan las neuronas en una RN y existen dos tipos fundamentales Redes Estáticas (Feed Forward), (RNFF) y Redes Recurrentes (RNR) o Redes Dinámicas.

En las RNFF, todas las señales neuronales se propagan hacia adelante. No existen conexiones hacia atrás y normalmente tampoco autorrecurrentes, ni laterales y son especialmente útiles en tareas de reconocimiento o clasificación de patrones [10].

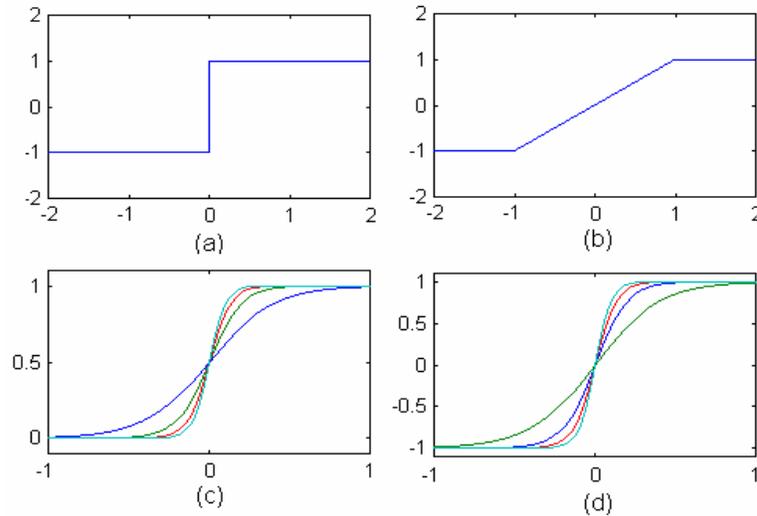


Figura 2.2: a) Escalón, b) Saturación c) Sigmoide d) Tangente hiperbólica

Las RNR son aquellas donde la información circula tanto hacia adelante como hacia atrás durante el funcionamiento de la red. Para que esto sea posible, existen conexiones hacia adelante y retroalimentaciones, [10]. En este trabajo de tesis utilizaremos una red híbrida que posee las dos características pero asociada principalmente con las RNR.

## 2.2. Lógica difusa

En los años 60's fue el comienzo de la teoría de la lógica difusa, esto fue debido por Lotfi A. Zadeh in 1965 con su publicación de "conjuntos difusos", en los 70's la teoría continuo creciendo, surgieron las aplicaciones reales, es justo decir que la teoría difuso establecida como el campo independiente el gran parte debido a la dedicación y notablemente al trabajo de Zadeh, lo más fundamental de la teoría de lógica difusa fue propuesto por Zadeh en los 60's y cerca de los 70's, después el propuso los algoritmos difusos en 1968, [12].

La lógica difusa se utiliza en un amplio rango de aplicaciones como:

Controladores de lavadoras, sistemas de aire acondicionado, video cámaras, mecanismos

de control, sistemas de control de mini submarinos, controladores del metro así como horno de microondas.

La lógica difusa permite tratar información imprecisa, en términos de conjuntos difusos veremos que estos conjuntos se combinan en reglas para definir acciones, por ejemplo, si la temperatura es alta entonces enfría mucho. De esta manera, los sistemas de control basados en lógica difusa combinan una variable de entrada (definidos en términos de conjuntos difusos), por grupos que producen uno o varios valores de salida. Hablando ya en términos más rigurosos, la teoría de lógica difusa parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia al conjunto, definida ésta como un número real entre 0 y 1, así se introduce el concepto de lógica difusa determinado a un valor lingüístico. Para cada conjunto o subconjunto difuso se define una función de pertenencia o inclusión  $\mu_A(t)$ , que indica el grado en el cual la variable  $t$  está incluida en el concepto que esta representado por la etiqueta A.

### 2.2.1. Conjuntos difusos

Recordando el ejemplo de las personas, esta situación puede describirse asignando un 1 a todos los elementos incluidos en el conjunto y un 0 a los no incluidos. A la función que asigna estos valores se llama función de inclusión o pertenencia (membership function).

A los subconjuntos difusos se les puede aplicar determinados operadores, o bien pueden realizarse operaciones entre ellos. Al aplicar un operador sobre un sólo conjunto difuso se obtiene otro conjunto difuso, de la misma manera al combinar dos o más subconjuntos mediante algunas operaciones, se obtendrá otro conjunto.

Sean los subconjuntos difusos identificados por las etiquetas  $A$  y  $B$ , asociados a una variable lingüística  $x$ , entonces se pueden definir tres operaciones básicas: complemento unión e intersección. Estas operaciones pueden expresarse de la siguiente manera en términos de funciones de pertenencia de los conjuntos difusos  $A$  y  $B$ .

Sea  $U$  un conjunto cualquiera y  $M = [0, 1]$ , su conjunto asociado de pertenencia. Si se considera a un conjunto difuso  $A \in U$ , entonces el complemento de A será:

- Complemento  $\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in U$

Sea  $U$  un conjunto cualquiera y  $M = [0, 1]$ , su conjunto asociado de pertenencia. La unión difusa de dos conjuntos difusos,  $A \in U$  y  $B \in U$ , se define como:

- Unión  $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \equiv \mu_A(x) \vee \mu_B(x) \quad \forall x \in U$ , donde  $\vee$  indica la operación máxima.

Sea  $U$  un conjunto cualquiera y  $M = [0, 1]$ , su conjunto asociado de pertenencia. La intersección difusa de dos conjuntos difusos,  $A \in U$  y  $B \in U$ , se define como:

- Intersección  $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \equiv \mu_A(x) \wedge \mu_B(x) \quad \forall x \in U$ , donde  $\wedge$  indica la operación mínima.

Es importante resaltar que el funcionamiento de estas operaciones básicas coincide con los correspondientes a los de la teoría clásica de conjuntos.

#### Operaciones entre conjuntos difusos

	<i>Operación</i>	<i>Rango</i>	
Igualdad	$\mu_A(x) = \mu_B(x)$	$x \in U$	
Unión	$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$	$\forall x \in U$	(2.8)
Intersección	$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$	$\forall x \in U$	
Complemento	$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$	$x \in U$	
Norma	$\mu_{Norma(A)}(x) = \frac{\mu_A(x)}{\max[\mu_A(x)]}$	$x \in U$	
Concentración	$\mu_{Conc(A)}(x) = (\mu_A(x))^2$	$x \in U$	
Dilatación	$\mu_{Dilat(A)}(x) = (\mu_A(x))^{0,5}$	$x \in U$	

La arquitectura típica de un control difuso lógico (FCL) se muestra en la Figura (2.3). El cual se compone de cuatro principales componentes: un borrosificador, una regla difusa base, una maquina de inferencia (Inference Engine) y un desborrosificador (defuzzifier). En la salida del desborrosificador no hay una acción de control para una planta, entonces el sistema es un sistema de decisión de lógica difusa. El borrosificador tiene el efecto de transformar los datos medidos alterados.

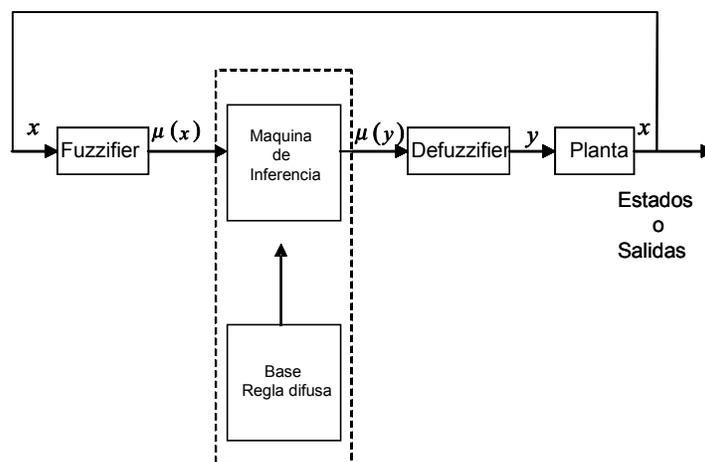


Figura 2.3: Arquitectura básica de un controlador de lógica difusa (FLC)

### 2.2.2. Borrosificador (Fuzzifier)

A continuación de enuncian algunas definiciones necesarias:

**Inferencia borrosa** también como en el caso de la lógica clásica, la borrosa se ocupa del razonamiento formal con proposiciones, pero a diferencia de ésta, los valores de las proposiciones pueden tomar valores intermedios entre verdadero y falso. De esta forma, los conjuntos borrosos también representan predicados en la lógica proposicional. El objeto de la lógica borrosa es proporcionar un soporte formal al razonamiento basado en el lenguaje natural, que se caracteriza por tratarse de un razonamiento de tipo aproximado, que hace uso de unas proposiciones que a su vez expresan información de carácter impreciso.

**Borrosificador** esto establece una relación entre puntos de entrada no borrosos al sistema  $x = (x_1, \dots, x_n)^T$ , y sus correspondientes conjuntos borrosos  $A$  en  $U$ . Se pueden utilizar diversas estrategias de borrosificación.

*Borrosificador singleton.* Es el método de borrosificación más utilizado, principalmente en sistemas de control consiste en considerar los valores propios discretos como conjuntos borrosos. De otra manera para cada valor de entrada  $x$  se define un conjunto  $A'$  que lo soporta, con una función de pertenencia  $\mu_A(x')$ , de modo que  $\mu_A(x) = 1$ , ( $x' = x$ ), y  $\mu_A(x') = 0$ ,

para todos los otros  $x' \in U$  en los que  $x' \neq x$ .

*Borrosificador no singleton.* Este método de borrosificador utiliza una función exponencial del tipo siguiente:

$$\mu_{A'}(x') = a * \exp\left[-\left(\frac{x' - x}{\sigma}\right)\right] \quad (2.9)$$

Función con forma de campana, centrada en el valor  $x$  de entrada, de anchura  $\sigma$  y amplitud  $a$ .

El conjunto de declaraciones consta de una regla base difusa, el cual es una parte vital de un sistema de lógica difusa (FLS), como se observa en la Figura siguiente (2.4), la base de la regla difusa (fuzzy inference engine), combina las declaraciones de la regla base, acordándose de aproximar por razón de la teoría a producir un mapeo del conjunto difuso en la entrada del espacio  $U$  a el conjunto difuso en la salida del espacio  $V$ . El borrosificador (fuzzifier), mapea en conjunto de entradas difusa en la entrada del espacio y el desborrosificador (defuzzifier), mapea agregando la salida del conjunto difuso una simple punto en la cresta de la salida en el espacio.

### 2.2.3. Desborrosificador (Defuzzifier)

Se llaman *dispositivos de inferencia borrosa* a los sistemas que interpretan las reglas de tipo IF - THEN de una base de reglas, con el fin de obtener los valores de salida a partir de los actuales valores de las variables lingüísticas de entrada al sistema.

**Desborrosificador** es la función que transforma un conjunto borroso en  $V$ , normalmente la salida de un dispositivo de inferencia borrosa, en un valor no borroso  $y \in V$ . Para esto se utilizan diversos métodos.

*Desborrosificador por máximo*, esta definido como:

$$y = \arg \sup_{y \in V} (\mu_{B'}(y)) \quad (2.10)$$

es decir,  $y$  es el punto de  $V$  en que  $\mu_{B'}(y)$  alcanza su valor máximo, donde  $\mu_{B'}(y)$  está definido según la ecuación

$$\mu_{B'}(y) = \mu_{B^1}(y) \dot{+} \dots \dot{+} \mu_{B^M}(y) \quad (2.11)$$

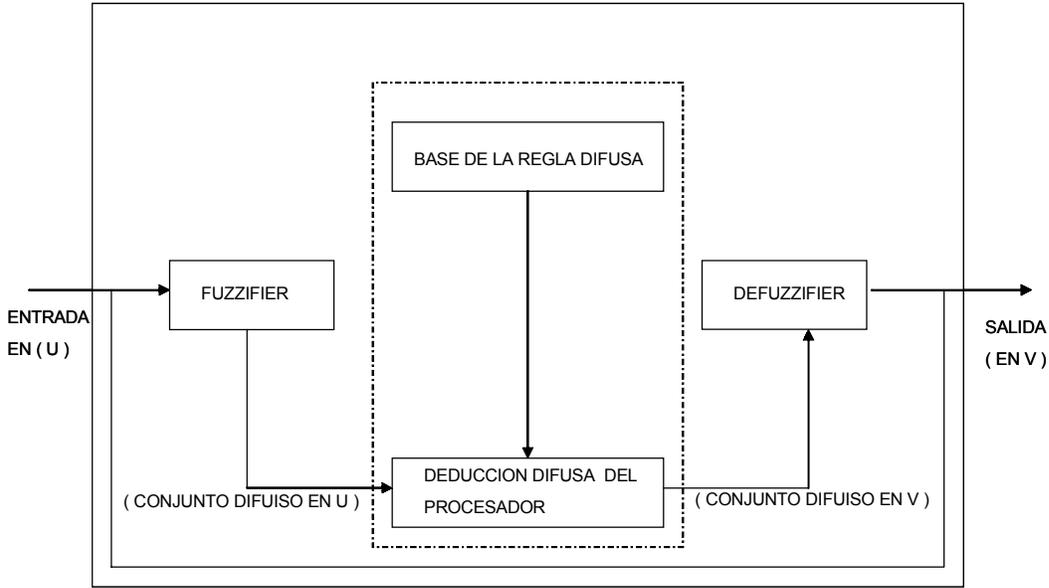


Figura 2.4: Estructura se un sistema de lógica difusa

*Desborrosificador por medida de centros*, de la manera siguiente

$$y = \frac{\sum_{l=1}^M \bar{y}^l (\mu_{B'}(\bar{y}^l))}{\sum_{l=1}^M (\mu_{B'}(\bar{y}^l))} \quad (2.12)$$

donde  $\bar{y}^l$  representa el centro del conjunto fuzzifier  $G^l$  ( definido como el punto de  $V$  en el que  $\mu_{G^l}(y)$  alcanza su valor máximo ), y  $\mu_{B'}(y)$  esta definido como:

$$\mu_{B'}(y) = \sup_{x \in U} \left[ \mu_{F_1^l x \dots F_n^l \rightarrow G^l}(\mathbf{x}, y) * \mu_{A'}(x) \right] \quad (2.13)$$

*Desborrosificador por centro de área*, definido como, [11].

$$y = \frac{\sum_{l=1}^M M^l (\mu_{B'}(\bar{y}^l))}{\sum_{l=1}^M A^l (\mu_{B'}(\bar{y}^l))} = \frac{\sum_{l=1}^M \int_V \mu_{B'}(\bar{y}^l)^2 dy M^l (\mu_{B'}(\bar{y}^l))}{\sum_{l=1}^M \int_V \mu_{B'}(\bar{y}^l) dy} \quad (2.14)$$

donde  $M^l$  es el momento (entorno al eje y del universo de discurso de la salida  $V$ ), de la función de inclusión del conjunto borroso  $G^l$ ,  $A^l$  es el área, y  $\mu_{B'}(y)$  está definida como la ecuación (2.13).

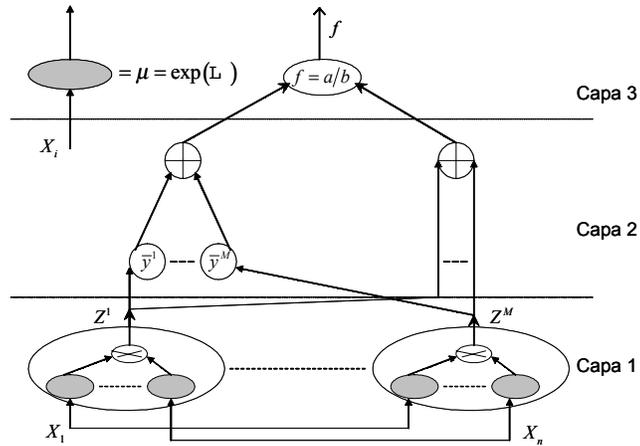


Figura 2.5: Representación de la red de un sistema difuso con una red neuronal de tres capas

La Figura (2.5), representa la salida del sistema difuso, es calculado como  $f(x) = a/b$ . Estas operaciones de tres etapas se muestra como la Figura siguiente con tres capas de la red de retroalimentación (feedforward).

### 2.3. Neuro difuso (Neurofuzzy)

El conjunto difuso o neurofuzzy son sistemas que usan lógica difusa que se comprende de juegos difusos y las reglas difusas que es combinando los datos numéricos y lingüísticos.

La lógica difusa usa las condiciones del idioma naturales como: frío, caluroso, caliente, pequeño, medio, grande. Tales condiciones no son precisas y no pueden representarse en la teoría normal del juego. Los juegos difuso les permiten a los miembros ser bien, la función parcial de pertenencia como el número de miembros multifijo normal. Este conjunto parcial habilita; La diferencia primaria entre los sistemas difuso y neurofuzzy, es que los sistemas difusos son especialistas en la construcción y los sistemas neurofuzzy son entrenados de datos. Aunque el software permite usar una combinación de los dos acercamientos interactivamente.

Las reglas de lógica difusa se usan en sistemas difusos y neuro difuso, representándose en términos lingüísticos la relación de entradas del sistema a sus salidas.

### 2.3.1. Estática y dinámica en sistemas difusos

Uno de los más importantes conceptos de propiedades de sistemas dinámicos es la estabilidad, sin embargo es difícil analizar la estabilidad de sistemas no lineales nos gustaría que los sistemas difusos y las RN, como a continuación se dice que el análisis de estabilidad de sistemas de controles difusos, el control difuso ha sido exitosamente aplicado en muchas aplicaciones industriales practicas, varias metodologías existentes para el análisis de estabilidad para sistemas de control difuso fueron analizados por: [Mamdani, 1976; Kickert and Mamdani, 1978; Tong, Chen 1989].

$$R^i : \text{Si } x \text{ es } A_i \ \& \ y \text{ es } B_i, \text{ entonces } z = a_0^i + a_1^i x + \dots a_n^i y \quad (2.15)$$

Donde  $x$ ,  $y$  son variables lingüísticas estas representan la distancia y orientación en relación a las cotas de la orientación,  $z$  es el próximo que sigue el ángulo decidido por el  $i$ th regla de control, y  $a_0^i$ ,  $a_n^i$  son los parámetros de entrada en el proceso de la identificación. La identificación se hace optimizando un mínimo cuadrados desempeñando el índice como un método de regresión lineal con pesos (Un algoritmo recursivo es el algoritmo de mínimos cuadrados) [Tong, 1978a; Takagi y Sugeno, 1983, 1985; Sugeno y Kang, 1986,1988]. Sumando el modelo de una acción de operador, este método es también usando el modelo (Identificación) controlando el proceso recordando sus datos entradas - salidas, el cual incluye parámetros de aprendizaje como la estructura de aprendizaje [Sugeno y Tanaka, 1991]. Estos son llamados reglas de control lingüísticas aproximándose al modelado difuso o identificación difusa. El modelo Takagi y Sugeno tiene la siguiente forma:

$$\begin{aligned} L^i : \text{Si } x(k) \text{ es } A_1^i \ \& \dots \ \& \ x(k-n+1) \text{ es } A_n^i \\ \ \& \ u(k) \text{ es } B_1^i \ \& \dots \ \& \ u(k-m+1) \text{ es } B_m^i \\ \text{Entonces } x^i(k+1) &= a_0^i + a_1^i x(k) + \dots a_n^i x(k-n+1) \\ &+ b_1^i u(k) + \dots b_m^i u(k-m+1) \end{aligned} \quad (2.16)$$

donde  $x(\cdot)$  es la variable de estado,  $x^i(k+1)$  es la salida de la regla  $L^i$ ,  $u(\cdot)$  es la variable de entrada

### 2.3.2. Comparaciones generales de sistemas difuso y redes neuronales

Los sistemas Difusos y las RN (Redes Neuronales), ambos son estimados modelos numéricos y sistemas dinámicos. Estos comparten la capacidad de mejorar la inteligencia del sistema trabajando en un indeterminado, impreciso, y el ruido del medio ambiente. Los sistemas difuso y las RN calculan ejemplos de funciones y se comportan tal como memorias asociadas. Ambos tienen una ventaja en lo tradicional de la estimación estadística y en la aproximación del control adaptable de una función estimación. Estos estiman una función sin necesidad de una descripción de la función matemática de donde la salida de la funcionalidad depende en la entrada; que ellos aprenden de ejemplos numéricos, la lógica difusa y la aproximación de neuronas son numéricamente por naturaleza.

A pesar de que los sistemas difusos y las RN son de forma similar, hay significantes diferencias entre estos. Estos difieren de la manera en que ellos estiman las funciones probadas, representan y dar un reconocimiento de la estructura, y asociativamente la deducción o mapeo de entradas y salidas.

Las RN tienen un grande numero favorablemente de elementos a procesar (nodos), el cual demuestra la habilidad de aprendizaje y generalización de patrones de entrenamiento [14], los sistemas difusos la base de sus decisiones de su entrada en la forma linguistica las variables se derivan de funciones el cual son formulas usadas para determinar el conjunto difuso el cual el valor pertenece y el grado de la función esta en el conjunto. Las variables entonces se igualan con las precondiciones de la regla lingüísticas Si Entonces (IF - THEN) (regla de lógica difusa), la respuesta de cada regla se obtiene a través de una implicación difusa.

Básicamente los sistemas difuso estiman funciones con conjuntos difusos por ejemplo  $(A_i, B_i)$ , mientras las RN usan puntos numéricos por ejemplo  $(\mathbf{x}_i, \mathbf{y}_i)$ , en una RN  $\mathbf{x}_i$  quizá representa los píxeles que hagan una imagen, mientras que  $\mathbf{y}_i$  quizá representan una clasificación de un vector, en una RN  $\mathbf{x}_i, \mathbf{y}_i$  dependiendo de sus entradas y salidas pueden aprender de un mapeo ajustando sus pesos, por otra parte los conjuntos difusos si  $A_i$  se caracteriza

el estado de un factor y  $B_i$  la posibilidad de la presencia o ausencia de varias rupturas en el factor, entonces las reglas difusas y la deducción se usaron para determinar la probabilidad que uno de estas rupturas quizá se represente por una función de  $A_i$ .

**Opción de sistemas difusos o RN.** Se entiende que una RN y la lógica difusa se usan cuando uno o más variables de estado son continuas y cuando un modelo matemático del proceso no existe o existe pero es tan difícil integrar es decir es tan complejo ser evaluado rápidamente para una operación en tiempo real. Entonces necesitamos encontrar cual método es el mejor y cuando debe ser usado, la teoría de las RN, conjuntos difusos y lógica difusa como desarrollando son sistemas abstractos que no hacen extensa demanda con respecto a biológicas o psicológicas recomendables.

Introducimos el teorema de estabilidad para sistemas difusos que proveer una condición suficiente que garantiza la estabilidad de un sistema difuso acordándose de la definición de estabilidad en el sentido de Lyapunov.

## 2.4. La estabilidad de la entrada a estado (ISS) y funciones de Lyapunov ISS

En esta sección estudiamos ISS, para sistemas no lineales en tiempo discreto, esto fue propuesto inicialmente por Sontang (1989, 1990), la estabilidad de la entrada a estado es otra elegante aproximación para analizar la estabilidad además el método de Lyapunov. Puede llevar a las conclusiones generales en la estabilidad usando la entrada y las características del estado, entendiendo las dependencias de las trayectorias de estados en magnitud de entrada para sistemas del tipo:

Considere el sistema no lineal de tiempo discreto siguiente:

$$\begin{aligned}x(k+1) &= f[x(k), u(k)] \\ y(k) &= h[x(k)]\end{aligned}\tag{2.17}$$

dónde  $u(k) \in \mathbb{R}^m$  es el vector de entrada,  $x(k) \in \mathbb{R}^n$  un vector de estado, y  $Y(k) \in \mathbb{R}^m$  es

la salida del vector  $f$  y  $h$  son generalmente funciones suaves no lineales,  $f, h \in C^\infty$ . De las siguientes definiciones tenemos que.

**Definición 2.1** *El sistema (2.17), se dice que es globalmente estable entrada estado ISS si ahí existe una función  $k$   $\gamma(\cdot)$  (continuo y estrictamente creciente si  $\gamma(0) = 0$ ) y una función  $k$   $\beta(\cdot)$  (función  $k$  y el  $\lim_{s_k \rightarrow \infty} \beta(s_k) = 0$ ), tal que para cada  $u \in L_\infty$ , entonces el  $\sup \{\|u(k)\|\} < \infty$ , y cada estado inicial  $x^o \in \mathbb{R}^n$ , esto dice que  $\|x(k, x^o, u(k))\| \leq \beta(\|x^o, k\|) + \gamma(u(k))$ .*

**Definición 2.2** *Una función continua  $V : \mathbb{R}^n \rightarrow \mathbb{R} \geq 0$  se llama una función suave de Lyapunov ISS del sistema (2.17), si: (a) existe un función  $k_\infty$  (función  $k_\infty$  y el  $\lim_{s_k \rightarrow \infty} \alpha_i(s_k) = \infty$ )  $\alpha_1(\cdot)$  y  $\alpha_2(\cdot)$  son tal que:*

$$\alpha_1(s) \leq V(s) \leq \alpha_2(s), \forall s \in \mathbb{R}^n$$

*(b) Ahí existe una función  $k_\infty$   $\alpha_3(\cdot)$  y una función  $k$   $\alpha_4(\cdot)$  tal que  $V_{k+1} - V_k \leq -\alpha_3(\|x(k)\|) + \alpha_4\|u(k)\|$ ,  $\forall x(k) \in \mathbb{R}^n$ ,  $u(k) \in \mathbb{R}^m$ .*

**Teorema 2.3** *Para un sistema no lineal de tiempo discreto ecuación (2.17), lo siguiente es equivalente [23].*

1. Es la estabilidad de entrada a estado (ISS).
2. Es estable y robusto.
3. Admite una función suave de Lyapunov ISS

Si un sistema no lineal es estable entrada a estado, el comportamiento del sistema, permanece acotado cuando sus entradas son acotadas.

**Propiedad 2.4** *Si un sistema no lineal es estable entrada a estado, el comportamiento del sistema, permanece acotado cuando sus entradas son acotadas.*



# Capítulo 3

## Estática neuro difuso para identificación

### 3.1. Estática difusa redes neuronales

En general, las redes neuronales difusa no se pueden igualar con precisión a los sistemas no lineales. El modelo dinámico puede ser el primero en llevar la tendencia de los parámetros al igual que el problema de inestabilidad. Algunas modificaciones robustas tienen que ser incluidas para garantizar la estabilidad de Lyapunov. En este modelo la estabilidad de entrada a estado se aproxima y se aplica para acceder al algoritmo de entrenamiento robusto de redes neuronales difusas. Concluimos que la ley descendiente del gradiente normal dependiente con una proporción de aprendizaje de tiempo variante es estable en el sentido  $L_\infty$ . El modelado neuronal difuso se aproxima a lo propuesto en este modelo, no necesita ninguna modificación robusta y es robusto en alguna inestabilidad acotada.

Ambas redes neuronales y la lógica difusas son consistentes universales, estos pueden aproximar cualquier función no lineal en alguna exactitud con tal de que las neuronas ocultas suficientes o reglas difusas que están disponibles. Recientemente resultados dicen que la fusión de las dos diferentes tecnologías parece ser muy eficaz para modelar una clase extensamente de sistemas no lineales complejos cuando nosotros no tengamos la información completa de

algún modelo [13],[17],[14]. El algoritmo de retroalimentación (*backpropagation*, BP), son siempre usados para conectar los parámetros de funciones del conjunto difuso (fuzzy sets) y los pesos del borrosificador (las redes neuronales). La baja velocidad de aprendizaje y el mínimo local son principales conflictos del algoritmo de BP [25]. Algunas modificaciones de aprendizaje del BP se deducen. [16] propuso un BP robusto que aprende para resistir el efecto del ruido y rechazar los errores a su totalidad durante la aproximación. [34] *B-spline* para funciones de conjunto difuso a minimizar la función del objeto robusto, la velocidad de convergencia fue mejorado. Las redes neuronales de RBF y la idea dinámica fueron aplicadas en [29], para determinar la estructura y parámetros del sistema neuronal difuso.

Para los ingenieros es muy importante asegurar la estabilidad en la teoría, antes de que ellos quieran aplicar el modelado a la técnica neuronal difusa a sistemas reales. Se conoce bien que los algoritmos normales de identificación (por ejemplo, gradiente y mínimo cuadrado) son estable para condiciones ideales.

Sin la seguridad del modelo dinámico, estos procedimientos adaptables pueden ser fácilmente inestables. La falta de robustez de identificación de parámetros se demostró en [18] y llegó a ser un problema directo en 80's. Varias técnicas de modificación robustas para identificación adaptable se propusieron en [21]. La actualización de los pesos del modelado neuro difuso es una clasificación de identificación de parámetros, el aprendizaje del gradiente descendiente normal y la retroalimentación son algoritmos estables, si el modelo neuronal difuso puede igualar exactamente a la planta no lineal. Generalmente tenemos que hacer algunas modificaciones de estos algoritmos por eso estos procesos de aprendizajes son estables.

El operador de proyección es muy efectivo para asegurar las cotas de los parámetros para el modelado difuso [33]. La técnica de la proyección también se usa por muchos sistemas difusos neuronales [31].

Otro método generalizado es usando las técnicas de modificación robustas de control adaptable [21], en modelado de redes difusas. Por ejemplo, [35] aplicando un cambio de modificaciones  $\sigma$  para prevenir la tendencia de parámetros.

El modelado neuronal difusa esta en el sentido de aproximación de una de caja negra. Todas las incertidumbres pueden ser considerando las partes de la caja negra, entonces, sin la

dinámica del modelo puede ser considerado dentro del modelo, no como las incertidumbres estructuradas. Por lo tanto las técnicas usadas comúnmente no son necesarias. En [24], los autores propusieron un algoritmo estable y un coeficiente de aprendizaje óptimo sin la modificación robusta. El coeficiente óptimo fue difícil ser encontrado, de tal manera que la búsqueda genética del algoritmo fue usado. Usando la teoría de pasividad, nosotros demostramos ese descenso del gradiente de los algoritmos sin la modificación robusta fueron estables y para algunas inestabilidad acotado para las redes neuronales recurrentes [41] neuro identificación [42].

En este modelo nosotros extendemos nuestros resultados anteriores para modelado de redes neuronales difusos. La estabilidad de la entrada a estado (ISS) la aproximación se aplica para obtener nuevas leyes de aprendizaje de modelado neuronal difuso. Para mejorar nuestro conocimiento, ISS se acercan para el sistema neuronal difuso no se ha establecido todavía en la literatura. Comparado con [24], nosotros también usamos el coeficiente de aprendizaje tiempo variante, pero este coeficiente de aprendizaje puede calcularse directamente de los datos de entrada salida. El algoritmo de aprendizaje es más simple que en [24].

Ambas Redes neuronales y la lógica difusa son los evaluaciones universales, ellos pueden aproximar cualquier función no lineal a alguna corrección, con tal de que las neuronas ocultas suficientes y las reglas difusas están disponibles. Escribiendo resultados muestra que el procedimiento de fusión de estas dos tecnologías diferentes parece ser muy eficaz para la identificación de sistemas no lineales [13], [17], [14]. El gradiente descendiente y la retroalimentación son siempre usados para ajustar los parámetros de las funciones de pertenencia (*fuzzy sets*) y los pesos de los borrosificadores (Redes Neuronales) para redes neuronales difusas. La lenta convergencia y el mínimo local son principales inconvenientes de estos algoritmos [25]. Algunas modificaciones fueron derivados en recientes publicaciones. [16] hacer pensar que la ley de retroalimentación robusta para resistir el efecto del ruido y rechazar los errores durante la tendencia de la aproximación. [34] usando la función de pertenencia *B-spline* para minimizar una función robusta, sus algoritmos pueden mejorar la velocidad de convergencia. En [29], las redes neuronales con funciones radiales básicas (RBF) son aplicados a sistemas difusos, las redes neuronales se aplicaron a sistemas difusos, un nuevo

acercamiento para determinar la estructura y parámetros de los sistemas neurales difusos fue propuesto.

El problema de estabilidad de identificación neural difusa es muy importante en las aplicaciones. Se conoce bien que los algoritmos de identificación normal (por ejemplo, gradiente descendiente de pendiente y mínimos cuadrados) son estable en las condiciones ideales. En la presencia de la dinámica del no modelado, ellos pueden volverse inestables. La falta de robustez de la identificación de los parámetros se demostró en [18] y llego a ser un problema directo en los 80's, cuando algunas técnicas de la modificación robustas fueron sugeridas [21]. El procedimiento de aprendizaje de las redes neural difusas puede considerarse como un tipo de identificación de parámetros. El gradiente descendiente dependiente y algoritmo de retroalimentación son estables, si los modelos neuronales difusos pueden igualar a las plantas no lineales exactamente. Sin embargo, algunas modificaciones robustas deben aplicarse para asegurar la estabilidad con respecto a las incertidumbres. La proyección del operador es una herramienta eficaz para garantizar la cota del modelado difuso [24]. Es también usado por muchos sistemas neurales difusos [31].

La identificación neuronal difusa usa datos de entrada salida y estructura del modelo. Puede considerarse como la aproximación de una caja negra. Todo las incertidumbres pueden ser consideradas como partes de la caja negra, es decir, la dinámica del no modelado está dentro del modelo de la caja negra, no como las incertidumbres estructuradas. Por consiguiente las técnicas de robusticidad comúnmente usadas no son necesarias. En [32], los autores sugirieron estabilidad y la velocidad de aprendizaje óptimo sin la modificación robusta, un algoritmo genético de investigación fue propuesto para encontrar el porcentaje óptimo. Sin embargo el algoritmo es complejo, y difícil de comprender. Usando la teoría de pasividad, nosotros demostramos con éxito que para tiempo continuo las redes neuronales recurrentes, los algoritmos de gradiente descendiente sin la modificación robusta eran estables y robustos en alguna cota [41], y para la identificación de tiempo continuo estos fueron también robustamente estables [42]. No obstante, hacer en tiempo discreto las redes neuronales difusa tienen características similares. El mejor conocimiento, ISS se aproxima para el sistema neural difuso todavía no se aplicó en la literatura.

En esta sección de estabilidad de entrada a estado (ISS) la aproximación se aplica a la identificación del sistema por redes neuronales difusas. Son considerados dos casos: 1) la proposición pertenencia son supuestamente conocidos, predeterminar de algún modo en progreso y aprendiendo a llevar sólo los parámetros a conclusión, y 2) la actualización de peso incluye ambos la proposición y los parámetros para concluir. Los nuevos algoritmos estables con las proporciones de aprendizaje de tiempo variantes se aplica a dos tipos de modelos neuronales difusos, llamados tradicionalmente modelo del tipo Mamdani y modelo de TSK.

El sistema no lineal para identificación

$$y(k) = h[x(k)] = \Psi[y(k-1), y(k-2), \dots, u(k-1), u(k-2), \dots] = \Psi[X(k)] \quad (3.1)$$

donde

$$X(k) = [y(k-1), y(k-2), \dots, u(k-d), u(k-d-1), \dots]^T \quad (3.2)$$

### 3.1.1. Estructura tipos mamdani estática difusa redes neuronales

Los modelos difusos Mandani y TSK se han usado por mas de dos decadas en el modelado de sistemas en la teoría de control (análisis y diseño).

El trabajo presenta un inovadora interpretación de los modelos difusos Mandani y TSK, estos modelos permiten saber que una mejor representación para sistemas dinámicos. El modelo Mandani es mejor para sistemas estáticos, mientras que el modelo TSk para dinámicos, El modelo Mandani sera una mejor aproximación para sistemas dinámico.

El sistema se representa por reglas difusas en la forma siguiente:

#### Modelo mandani

$$R^{(i_1, \dots, i_n)} : \text{si } (x_1 \text{ es } X_1^{(i_1)}) \text{ y, } \dots, \text{ y } (x_n \text{ es } X_n^{(i_n)}) \text{ entonces } (x_{n+1} \text{ es } X_{n+1}^{(i_1, \dots, i_n)}) \quad (3.3)$$

$$R^i : \text{si } x_1 \text{ es } A_{1i} \text{ y } x_2 \text{ es } A_{2i} \text{ y, } \dots, x_n \text{ es } A_{ni} \text{ entonces } \hat{y}_1 \text{ es } B_{1i} \text{ y, } \dots, \hat{y}_m \text{ es } B_{mi} \quad (3.4)$$

$\forall l = (1, 2, \dots, n)$  difuso si entonces las reglas para realizar un vector de un mapeo de entrada lingüística  $X = [x_1, \dots, x_n] \in \mathbb{R}^n$  a una variable de salida lingüística.  $A_{1i}, \dots, B_{mi}$  son conjuntos difusos estandares [33]. Para la variable de entrada  $x_i$  hay  $l_i$  conjunto difusos. En el caso de una conexión completa,  $l = l_1 \times l_2 \times \dots \times l_n$  de [33] sabemos que, usando la inferencia del producto, centro promedio y borrosificador singleton, el  $kth$  salida del sistema lógica difusa se expresa como:

$$\begin{aligned} \hat{y}_p &= \left( \sum_{i=1}^l w_{pi} \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] / \sum_{i=1}^l \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] \right) \\ &= \sum_{i=1}^l w_{pi} \phi_i \end{aligned} \quad (3.5)$$

donde  $\mu_{A_{ji}}$ , son las funciones de pertenencia del conjunto difuso  $A_{ji}$  y  $w_{pi}$  es el punto en el cual  $\mu_{B_{pi}} = 1$ . Si definimos

$$\phi_i = \prod_{j=1}^n \mu_{A_{ji}} / \sum_{i=1}^l \prod_{j=1}^n \mu_{A_{ji}}. \quad (3.6)$$

La ecuación (3.5) puede ser expresado como parámetros

$$\hat{Y}(k) = W_k \Phi [X(k)] \quad (3.7)$$

donde

$$\hat{Y}(k) = [\hat{y}_1, \dots, \hat{y}_m]^T = W_k = \begin{bmatrix} w_{11} & & w_{1l} \\ & \ddots & \\ w_{m1} & & w_{ml} \end{bmatrix} \quad (3.8)$$

el vector de datos  $\phi [X(k)] = [\phi_1, \dots, \phi_l]^T$ .

La arquitectura de una red neuronal difusa FNN ver Figura (3.1), consta de cuatro capas, los nodos de entradas y los nodos de salidas se encuentran en la capa I y la capa IV, las capas ocultas se forman de nodos de reglas para reglas difusas.

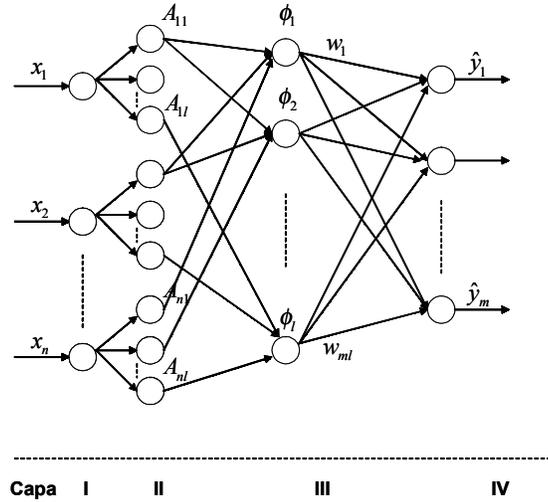


Figura 3.1: Arquitectura de un sistema neuronal difuso

### 3.1.2. Estructura tipos takagi sugeno kang estática difusa redes neuronales

Para el modelo neuronal difuso Takagi Sugeno Kang tenemos

$$R^i : \text{si } x_1 \text{ es } A_{1i} \text{ y } x_2 \text{ es } A_{2i} \text{ y, } \dots, x_n \text{ es } A_{ni} \text{ entonces } \hat{y}_j = p_{j0}^i + p_{j1}^i x_1 + \dots + p_{jn}^i x_n \quad (3.9)$$

donde  $j = 1, \dots, m$ . el  $p$ th salida del sistema de lógica difusa puede expresarse como

$$\hat{y}_p = \sum_{i=1}^l (p_{p0}^i + p_{p1}^i x_1 + \dots + p_{pn}^i x_n) \phi_i \quad (3.10)$$

donde  $\phi_i$  es definido como en la ecuación (3.6). La ecuación (3.10) puede ser expresado en la forma del tipo Mamdani (3.7),

$$\hat{Y}(k) = W(k) \Phi[x(k)]$$

donde  $\widehat{Y}(k) = [\widehat{y}_1 \cdots \widehat{y}_m]^T$

$$W(k) = \begin{bmatrix} p_{10}^1 \cdots p_{10}^l & p_{11}^1 \cdots p_{11}^l & \cdots & p_{1n}^1 \cdots p_{1n}^l \\ \vdots & \vdots & & \vdots \\ p_{m0}^1 \cdots p_{m0}^l & p_{m1}^1 \cdots p_{m1}^l & \cdots & p_{mn}^1 \cdots p_{mn}^l \end{bmatrix} \quad (3.11)$$

$$\Phi[x(k)] = \left[ \phi_1 \cdots \phi_l \quad x_1 \phi_1 \cdots x_1 \phi_l \quad \cdots \quad x_n \phi_1 \cdots x_n \phi_l \right]^T \quad (3.12)$$

Estructura Takagi Sugeno

## 3.2. Sistemas de identificación difusa

La idea principal es presentar una herramienta matemática para construir un sistema modelo difuso, generalmente hablando el modelo de construcción por datos de entrada y salida es caracterizado por dos cosas; el primero por una herramienta matemática para expresar un modelo de sistemas y el otro es un método de identificación.

También se muestra el método de identificación de un sistema usando datos de entrada y salida, como sabemos para la identificación de sistemas se divide en dos partes: estructura de identificación y parámetros de identificación.

La estructura de implicaciones difusa y algoritmos de razonamientos, denotamos la función de pertenencia de un conjunto difuso  $A$  como  $A(x)$ ,  $x \in X$ . Todos los conjuntos difusos son asociados con funciones lineales, entonces la función pertenencia es caracterizado por dos parámetros dando el grado mas alto 1 y el grado menos 0. El valor verdadero de una proposición " $x$  es  $A$  &  $y$  es  $B$ ", se expresa por:

$$|x \text{ es } A \ \& \ y \text{ es } B| = A(x) \wedge B(y) \quad (3.13)$$

Formas de implicaciones, la implicación difusa  $R$  es de la forma:

$$R : \text{es } f(x_1 \text{ es } A_1, \dots, x_k \text{ es } A_k) \text{ entonces } y = g(x_1, \dots, x_k) \quad (3.14)$$

donde

$y$	es la variable de respuesta cuyo valor es deducido
$x_1 - x_k$	proposición de variables que muestran en la parte del resultado
$A_1 - A_k$	conjunto difuso con funciones lineales representando un subespacio difuso en el cual la implicación $R$ puede ser aplicado
$f$	función lógica que conecta la proposición
$g$	función que implica el valor de $y$ cuando $x_1 - x_k$ satisface la proposición

En la proposición si  $A_i$  es igual a  $X_i$  para algún  $i$  donde  $X_i$  es el universo.

### Proposición de cambio de variables

Se propone un algoritmo para el cambio de variables de las variables de entrada considerables. Como han sido previamente declaradas, todas las variables de las consecuencias que no siempre aparecen en las proposiciones. Hay dos problemas interesados con el algoritmo. Uno es el cambio de variables: para elegir una variable en la proposición esto implica que es inciertamente dividido. El otro es el número de divisiones. El problema completo es una combinación, por eso en general ahí al parecer no teóricamente hay una aproximación disponible.

Aquí justamente se lleva a buscar un método descrito en los siguientes pasos.

Suponemos que construimos un modelo difuso de entrada  $K$  —ésima  $x_1, \dots, x_k$  y un sistema de salida única.

- El rango de  $x_1$  se divide en dos subespacios difusos "grande" y "pequeño", el rango de las otras variables  $x_2, \dots, x_k$  no son divididas, se refiere que solo  $x_1$  aparece en la proposición de la implicación. Entonces el modelo consiste de dos implicaciones.

Si  $x_1$  es *grande*<sub>1</sub> entonces ...

Si  $x_1$  es *chico*<sub>2</sub> entonces ...

Este modelo se llama *1-1*. Simultáneamente, un modelo en el cual el rango de  $x_2$  se divide y el rango de las otras variables  $x_1, x_3, \dots, x_k$  no son divididas entonces son el modelo *1-2*. De esta manera tenemos  $k$  modelos, en el cual se compone de dos implicaciones. En general, el modelo *1-i* es de la forma

Si  $x_i$  es *grande* <sub>$i$</sub>  entonces ...

Si  $x_i$  es *chico* <sub>$i$</sub>  entonces ...

- Para cada modelo los parámetros de la proposición óptima y parámetros de consecuencia son encontrados por el algoritmo que se describe en la sección previa. El modelo óptimo con el mínimo índice de desempeño se aprueba fuera del modelo  $k$ . Este paso se llama estado estable.
- Comenzando de un estado estable como el primer paso, se dice que el modelo  $1-i$ , donde solo la variable  $x_i$  se muestra en la proposición, toma todas las combinaciones de  $x_i - x_j$  ( $j = 1, 2, \dots, k$ ) y divide el rango de cada variable en dos subespacios difusos. Para la combinación de  $x_i - x_i$ , el rango de  $x_i$  se divide en cuatro subespacios, por ejemplo "grande", "medio grande", "medio chico" y "chico". Entonces se obtiene el modelo  $k$  el cual cada modelo se llama  $2 - j$ . Entonces cada modelo consiste de  $2 \times 2$  implicaciones. Encontramos de nuevo un modelo con el mínimo índice de proposición como el paso 2 que también se llama un estado estable en este paso.
- Repitiendo el paso 3 simultáneamente de manera de poner otra variable dentro de la proposición.
- La búsqueda se detiene si cualquiera de los siguientes criterios se satisfacen.
  - 1) El índice de la proposición de estado estable llega a ser menos que el valor predeterminado.
  - 2) El número de implicaciones de un estado estable excede los números predeterminados.
 El cambio de variables en la proposición se muestra en la Figura (3.2).

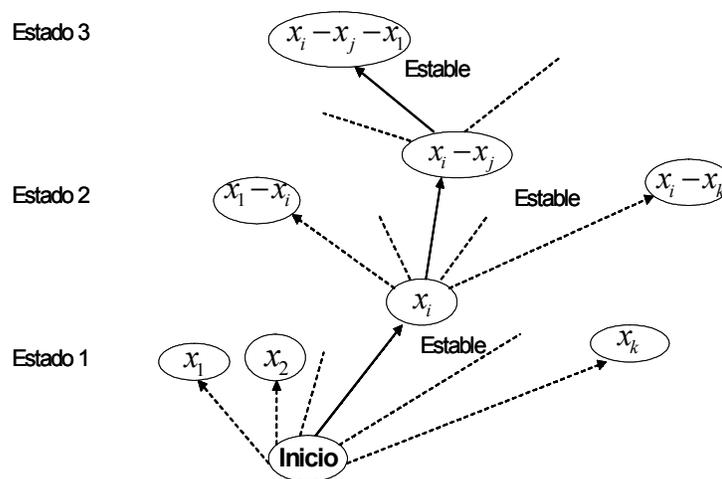


Figura 3.2: Selección de variables de proposición

### 3.3. Sistemas de identificación difusa neuronal

#### 3.3.1. Funciones pertenencia conocidas

Cuando tenemos alguna información anterior de la planta a identificar, podemos construir reglas difusas como en las ecuaciones (3.4) y (3.9). Suponemos que las funciones pertenencia  $A_{1i}, \dots, A_{ni}$  son dados por conocimiento anterior, entonces  $\phi_i = \prod_{j=1}^n \mu_{A_{ji}} / \sum_{i=1}^l \prod_{j=1}^n \mu_{A_{ji}}$  se conoce [16], [29], [35]. El objetivo del modelado de la neurona difusa es encontrar valores del centro de  $B_{1i} \cdots B_{mi}$  ( los pesos entre las capas III y IV como se muestran en la Figura (3.1)), tal que la salida  $\hat{Y}(k)$  de la red neuronal difusa (3.7), sigue la salida  $Y(k)$  de la planta no lineal (3.1).

El error de identificación se define como:

$$e(K) = \hat{Y}(K) - Y(K) \quad (3.15)$$

El modelado neuronal difuso discutido en este modelo es un tipo de identificación en línea, entonces usaremos el modelado del error  $e(k)$  para entrenar las redes neuronales difusas la ecuación (3.7) en línea tal que  $\hat{Y}(k)$  puede aproximarse a  $Y(k)$ . según las teorías de

aproximación de la función de lógica difusa [24] y las redes neuronales [9], el proceso de identificación no lineal la ecuación (3.1) se puede representar como:

$$Y(k) = W^* \Phi[X(k)] - \mu(k) \quad (3.16)$$

donde  $W^*$  son los pesos desconocidos que pueden minimizar el no modelado dinámico  $\mu(k)$ .

**Teorema 3.1** *Si nosotros asumimos que las redes neuronales difusas la ecuación (3.7), el modelado de la planta no lineal de la ecuación (3.1), el siguiente algoritmo de descenso de gradiente con una proporción de aprendizaje de tiempo variante se puede hacer la identificación del error acotado  $e(k)$ .*

$$W(k+1) = W(k) - \eta_k e(k) \Phi^T[X(k)] \quad (3.17)$$

donde  $\eta_k = \frac{\eta}{1 + \|\phi[X(k)]\|^2}$ ,  $0 < \eta \leq 1$ . El error de identificación normalizado satisface el siguiente promedio de desempeño

$$e_N^2(k) = \frac{e^2(k)}{1 + \max_k (\|\phi[X(k)]\|^2)} \quad (3.18)$$

$$J = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T e_N^2(k) \leq \bar{\mu} \quad (3.19)$$

donde  $\bar{\mu} = \max_k [\mu^2(k)]$ .

**Demostración.** [43] ■

**Propiedad 3.2** *Porque el modelo neuronal difuso no puede igualar los sistemas no lineales exactamente. Así nosotros no podemos hacer que los parámetros de convergencia neuronales difusas a sus valores óptimos, nosotros solo queremos forzar la salida de las redes neuronales para seguir la salida de la planta. Aunque los valores de los pesos no pueden converger a sus valores óptimos, la ecuación (3.19) las muestras de la identificación normalizado del error convergerá al radio de la bola  $\bar{\mu}$ .*

### 3.3.2. Funciones pertenencia desconocidas

Cuando consideramos la planta como una caja negra, ninguna de las proposiciones ni los parámetros consecuentes son conocidos (ver [28], [34], [32]). Ahora el objeto del modelado de la neurona difusa es encontrar valores centrales de  $B_{1i} \cdots B_{mi}$ , como serían las funciones pertenencia  $A_{1i} \cdots A_{ni}$ , tal que la red neuronal difusa (3.7), puede seguir la planta no lineal (3.1).

La función gaussiana pertenencia son aprovechados para identificar las reglas difusas en esta sección, el cual se define como

$$\mu_{A_{ji}} = \exp \left( -\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) \quad (3.20)$$

**Teorema 3.3** *Si nosotros usamos la red neuronal difusa de tipo Mamdani (3.4) para identificar la planta no lineal (3.1), el siguiente algoritmo de retroalimentación, hacer que la identificación del error  $e(k)$  sea acotado.*

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) Z(k)^T \\ c_{ji}(k+1) &= c_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} (\hat{y}_q - y_q) \\ \sigma_{ji}(k+1) &= \sigma_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{(x_j - c_{ji})^2}{\sigma_{ji}^3} (\hat{y}_q - y_q) \end{aligned} \quad (3.21)$$

donde  $\eta_k = \frac{\eta}{1 + \|Z\|^2 + 2\|D_z\|^2}$ ,  $0 < \eta \leq 1$ . El promedio de la identificación del error satisface

$$J = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T e^2(k) \leq \frac{\eta \bar{\zeta}}{\pi} \quad (3.22)$$

donde  $\pi = \frac{\eta}{(1 + \kappa)^2} > 0$ ,  $\kappa = \max_k (\|Z\|^2 + 2\|D_z\|^2)$ ,  $\bar{\zeta} = \max_k [\zeta^2(k)]$ .

**Demostración.** [43] ■

Para el modelo neuronal difuso del tipo *Takagi Sugeno Kang* ecuación (3.4) seleccionamos  $A_{ji}$ , como funciones gaussiana. La  $q$ th salida del sistema lógico difuso puede expresarse como

$$\hat{y}_q = \sum_{i=1}^l \left( \sum_{k=0}^n p_{qk}^i x_k \right) \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) / \left[ \sum_{i=1}^l \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) \right] \quad (3.23)$$

donde  $x_0 = 1$ . La parte  $\sum_{i=1}^l (w_{qi}^* - w_{qi}) z_i/b_q$  es la ecuación (??) fue cambiado como

$$\sum_{i=1}^l \left( \sum_{k=0}^n (p_{qk}^{i*} - p_{qk}^i) x_k \right) z_i/b \quad (3.24)$$

El teorema siguiente nos da un algoritmo estable para la red neuronal difusa del tipo TSK.

**Teorema 3.4** *Si usamos la red neuronal difusa de tipo TSK de la ecuación (3.23) para identificar la planta no lineal ecuación (3.1), el algoritmo siguiente hace la identificación del error acotado  $e(k)$ .*

$$\begin{aligned} p_{qk}^i(k+1) &= p_{qk}^i(k) - \eta_k (\hat{y}_q - y_q) \frac{z_i}{b} x_k \\ c_{ji}(k+1) &= c_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} (\hat{y}_q - y_q) \\ \sigma_{ji}(k+1) &= \sigma_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{(x_j - c_{ji})^2}{\sigma_{ji}^3} (\hat{y}_q - y_q) \end{aligned} \quad (3.25)$$

donde  $\eta_k = \frac{\eta}{1 + \|Z\|^2 + 2\|D_z\|^2}$ ,  $0 < \eta \leq 1$ .

**Demostración.** [43] ■

Uno puede observar que TSK necesita solo una formulación diferente comparativo al tradicional modelo Mamdani y la diferencia entre el teorema (3.3) y teorema (3.4), es contener de la ley de actualización para  $p_{qk}^i(k)$ .

## 3.4. Simulación

**Ejemplo 3.5** *R si  $x_1$  es chico y  $x_2$  es grande entonces*

$$y = x_1 + x_2 + 2x_3 \quad (3.26)$$

*La implicación de estado que si  $x_1$  es chico y  $x_2$  es grande, entonces el valor de  $y$  debería ser igual a la suma de  $x_1$ ,  $x_2$  y  $2x_3$ , donde  $x_3$  no es condicionado de la propuesta.*

*R : si  $x_1$  es  $A_1$  &...&  $x_k$  es  $A_k$ , entonces  $y = p_0 + p_1x_1 + \dots + p_kx_k$ .*

El algoritmo de razonamiento, suponemos que tenemos implicaciones  $R^i$  ( $i = 1, \dots, n$ ) anteriormente entonces tenemos:

$(x_1 = x_1^0, \dots, x_k = x_k^0)$ , donde  $x_1^0 - x_k^0$  son singleton, el valor de  $y$  se deduce en los pasos siguientes:

1.- Para cada implicación  $R^i, y^i$  se calcula por la función  $g^i$  el resultado es:

$$y^i = g^i(x_1^0, \dots, x_k^0) = p_0^i + p_1^i x_1^0, \dots, + p_k^i x_k^0 \quad (3.27)$$

2.- El valor verdadero de la proposición  $y = y^i$  se calcula por la ecuación:

$$|y = y^i| = |x_1^0 \text{ es } A_1^i \& \dots \& x_k^0 \text{ es } A_k^i| \quad (3.28)$$

$$\wedge |R^i| = (A_1^i(x_1^0) \wedge \dots \wedge A_k^i(x_k^0)) \wedge |R^i| \quad (3.29)$$

donde  $|*|$  se refiere al valor verdadero de la proposición  $*$  &  $\wedge$  es la proposición para el operador mínimo, y  $|x^0 \text{ es } A| = A(x^0)$ , entonces el grado de  $x^0$ . Por simplicidad se asume que  $|R^i| = 1$  por eso el valor verdadero del resultado se obtiene como:

$$|y = y^i| = A_1^i(x_1^0) \wedge \dots \wedge A_k^i(x_k^0) \quad (3.30)$$

3.- la salida final  $y$  se deduce de  $n$  implicaciones es dado como el promedio de todos los  $y^i$  con los pesos

$$|y = y^i| : y = \frac{\sum |y = y^i| \times y^i}{\sum |y = y^i|} \quad (3.31)$$

En este ejemplo se muestra la identificación usando un sistema de datos recopilados de entrada salida con ruidos. La deducción típica del ruido esta en el 5% de la salidas. Este sistema tiende a ser notado que se puede identificar justo los mismos parámetros de la condición como el sistema original si el ruido no existe.

Es de gran importancia un punto de salida anteriormente hecho, si este no es el caso se puede afirmar la valides de un algoritmo de identificación a la vez con una descripción de lenguaje sistema difuso.

Suponemos que el sistema original existe ver Figura (3.3), con las dos implicaciones siguientes:

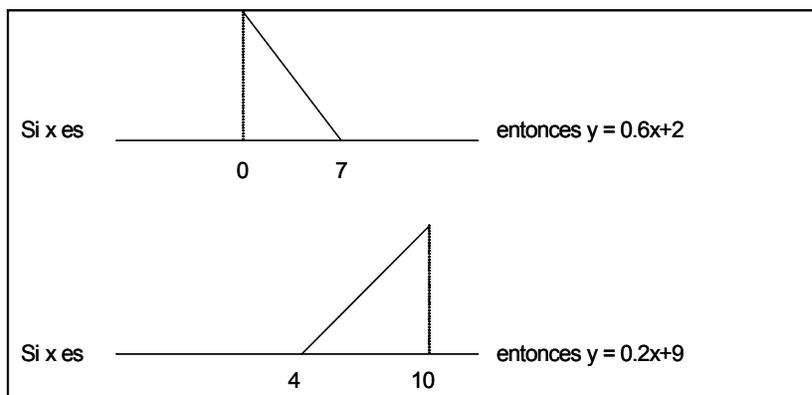


Figura 3.3: Sistema

la función es consecuencia de implicaciones en datos con ruidos de entrada salida como se muestra a continuación Figura (3.4), los parámetros propuestos de identificación como se muestran, podemos ver que casi siempre son los mismos parámetros y estos han sido derivados. En esta sección, se sugiere un algoritmo de aprendizaje estable que es evaluado como una función de aproximación y un sistema a identificar.

### Ejemplo 3.6 Una Función de aproximación de 2 dimensiones

Nosotros queremos usar el algoritmo estable para aproximar la función siguiente:

$$f(x_1, x_2) = 0.52 + 0.1x_1 + 0.28x_2 - 0.6x_1x_2$$

este ejemplo se toma de [33], ellos usan el siguiente sistema difuso para aproximar esto.

$$\hat{f} = \frac{\sum_{x_1} \sum_{x_2} f(x_1, x_2) \mu_{A_1} \mu_{A_2}}{\sum_{x_1} \sum_{x_2} \mu_{A_1} \mu_{A_2}} \quad (3.32)$$

se usara la red neuronal difusa la ecuación (3.7). La entrada  $X(k) = [x_1(k), x_2(k)]$ . Usamos el mismo conjunto difuso como en [33]. El numero de conjunto difuso para cada variable de entrada en la ecuación (3.5). Hay 25 reglas difusas. La función que asocia para  $x_1$  y  $x_2$  son funciones pertenencia en  $[2, 2]$ , ver Figura (3.5). como en la Figura (3.1),  $n = 2, l_1 = l_2 =$

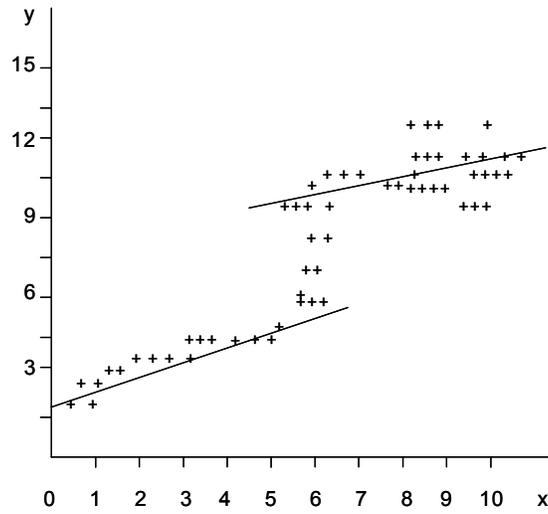


Figura 3.4: Datos de consecuencia y ruido

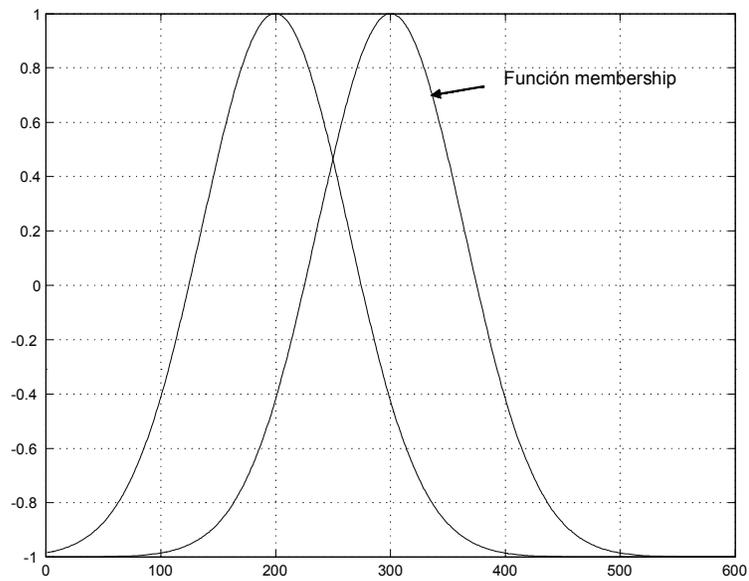


Figura 3.5: Función gaussiana

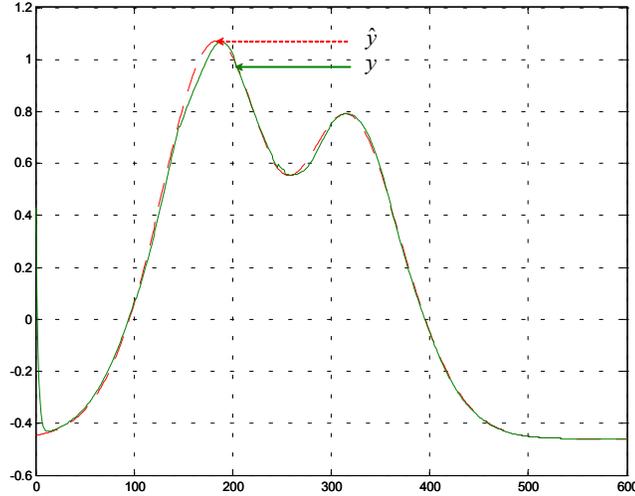


Figura 3.6: Aproximación de la función

11,  $l = 121$ ,  $m = 1$ . y  $x_1(k)$  y  $x_2(k)$  son seleccionados como:

$$\begin{aligned} x_1(k) &= \left[ 2 * \exp\left(\frac{-(x(k) - c)^2}{\sigma^2}\right) - 1 \right] \\ x_2(k) &= \left[ 2 * \exp\left(\frac{-(x(k) - 8.25)^2}{\sigma^2}\right) - 1 \right] \end{aligned} \quad (3.33)$$

donde  $k = 1, 2, \dots, T$ . En esta simulación nosotros dejamos  $T = 600$ . El algoritmo de aprendizaje es como en la ecuación (3.17), con  $\eta = 1$ . El resultado de identificación y los pesos se muestran en la Figura (3.6). Para verificar la efectividad del algoritmo, nosotros encontramos después de  $\eta \geq 2.7$ , el proceso de aprendizaje se hace inestable. El resultado de identificación con  $\eta = 2.7$ . se muestra en la Figura (3.7). El teorema (3.1), nos da una condición necesaria de  $\eta$  para aprendizaje estable, con  $\eta \leq 1$ . En este ejemplo la estabilidad se limita a ser aumentado a un valor de  $\eta < 2.7$ . Para una aplicación real, podemos seleccionar un  $\eta = 1$ . Del teorema (3.1) el modelado del error asegura que es estable. Si queremos un proceso de aprendizaje rápido aumentando un poco más el valor de  $\eta$  quizás también sea el error de

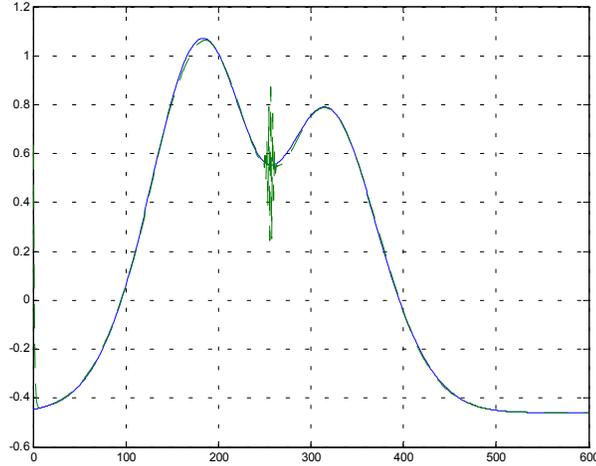


Figura 3.7: Aprendizaje inestable

identificación estable.

Para este sistema el error medio cuadrático queda expresado en la Figura (3.8).

### Ejemplo 3.7 Identificación del sistema no lineal

Usamos un sistema no lineal para ilustrar el aprendizaje del gradiente descendiente ecuación (3.17) y el algoritmo de retroalimentación ecuación (3.21) y el problema estable. La planta identificada es (Ejemplo 4 de [27]).

$$y(k) = \frac{y(k-1)y(k-2)y(k-3)u(k-1)[y(k-3)-1] + u(k-1)}{1 + y(k-2)^2 + y(k-3)^2} \quad (3.34)$$

la entrada de la señal se selecciona como:

$$u(k) = \begin{cases} \sin\left(\frac{\pi}{15}\right) & k < 200 \\ 1 & 200 \leq k < 400 \\ -1 & 400 \leq k < 600 \\ 0.3 \sin\left(\frac{\pi}{15}k\right) + 0.1 \sin\left(\frac{\pi}{22}k\right) + 0.603 \sin\left(\frac{\pi}{8.5}k\right) & 600 \leq k < 1000 \end{cases} \quad (3.35)$$

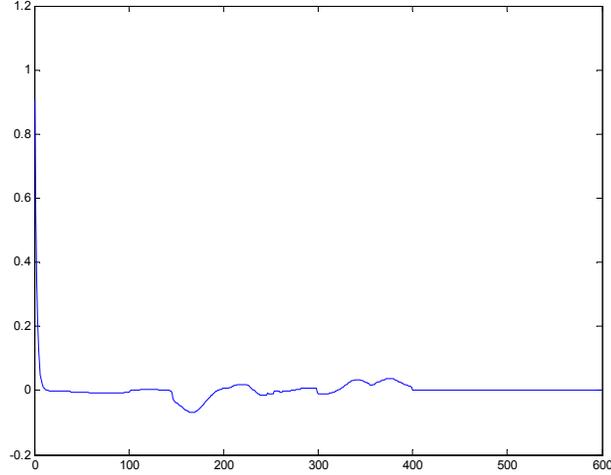


Figura 3.8: Error medio cuadrático

nosotros usamos la red neuronal difusa la ecuación (3.7), para identificar la ecuación (3.34),

$$X(K) = [y(k-1), y(k-2), y(k-3), u(k-1), u(k-2)]^T$$

para la Figura (3.1),  $n = 5$ ,  $l_j = 10$  ( $j = 1, \dots, 5$ ). El método para evitar el peso computacional, nosotros usamos una simple conexión, es decir, la entrada para  $\phi_k$  es solo  $A_{1k}$ ,  $A_{k2}$ , ...,  $A_{nk}$  ( $k = 1, \dots, l$ ). Así  $l = 10$ ,  $m = 1$ . Las funciones pertenencia son escogidas como funciones gaussianas.

$$\mu_{A_{ji}}(k) = \exp\left(-\left(\frac{x_j(k) - c_{ji}}{\sigma_{ji}}\right)^2\right) \quad (3.36)$$

El centro  $c_{ji}$  y el ancho  $\sigma_{ji}$ , son constantes aleatorias entre  $[0, 1]$ . La complejidad del modelo es importante en el contexto de la identificación del sistema, el cual corresponde al número de unidades de reglas del modelo neuronal difuso. En esta simulación nosotros intentamos probar diferentes números de nodos ocultos, nosotros encontramos que después que los dominios de los nodos es más que  $l > 10$ , la identificación aproximada no mejorara mucho. Nosotros seleccionamos  $\eta = 1$  para actualizar los pesos  $W_k$  como en la ecuación (3.7). Las condiciones

iniciales para  $W_k$  son aleatorios entre  $[0, 1]$ . El resultado de identificación se muestra en Figura (3.9), y el error cuadrático medio en la Figura (3.10).

Del teorema (3.1). Dar una condición necesaria de  $\eta$  para el aprendizaje estable,  $\eta \leq 1$ . En este ejemplo, nosotros encontramos que si  $\eta \geq 3.1$ , el proceso de aprendizaje llega a ser inestable.

Comparando el gradiente normal descendiente, el porcentaje de aprendizaje en tiempo variante  $\eta_k$  en la ecuación (3.17) es más fácil para ser comprendido. Este ejemplo se ha hecho en varios artículos. Por ejemplo, una red neuronal multicapas [27] (la velocidad de aprendizaje fijando  $\eta = 0.025$ ), la red difusa recurrente tipo TSK [5] y las redes neuronales difusas dinámica [6]. Uno puede ver que el algoritmo propuesto en este modelo la ecuación (3.17), es más fácil de comprender para el diseño. Quizás ahí existe una  $\eta$  mejor como en [24], pero muchas de las simulaciones han sido hechas para encontrar un porcentaje de aprendizaje mejor de  $\eta$ .

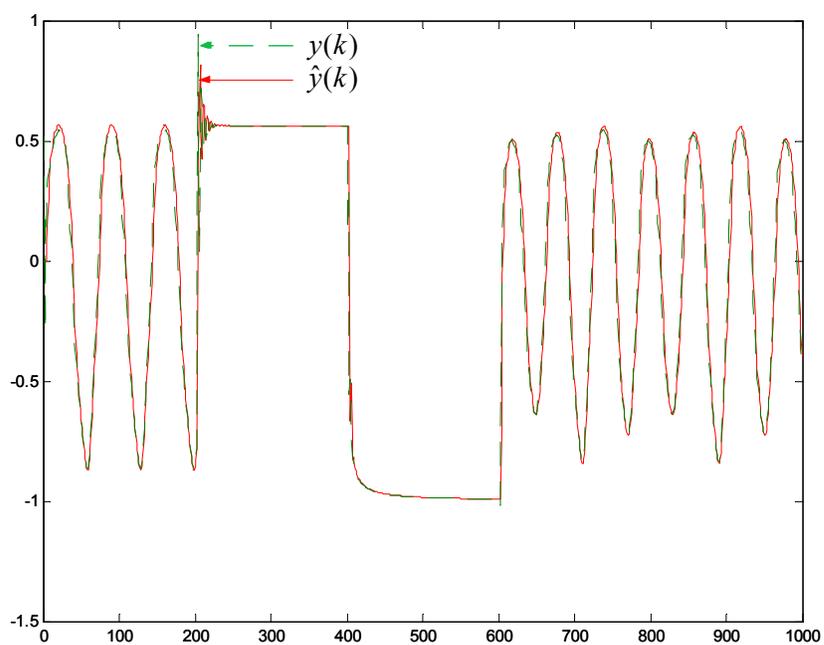


Figura 3.9: Modelado de un sistema no lineal

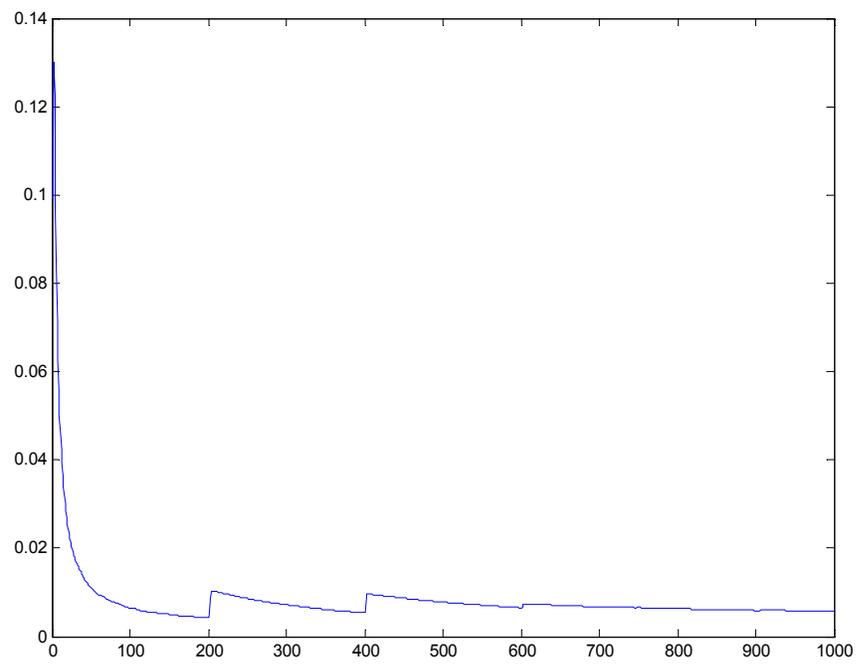


Figura 3.10: Error cuadrático medio



# Capítulo 4

## Dinámica neuro difuso para identificación

### 4.1. Dinámica difusa redes neuronales

#### 4.1.1. Estructura tipo mamdani dinámica difusa redes neuronales

Un modelo difuso de un grupo se presenta como una colección de reglas difusas de la forma siguiente:

$$\begin{aligned} R^i : & \text{ si } \hat{x}_1(k) \text{ es } A_{1i} \text{ y } \hat{x}_2(k) \text{ es } A_{2i} \text{ y, } \dots, \hat{x}_n(k) \text{ es } A_{ni} \\ & \text{ entonces } \hat{x}_1(k+1) \text{ es } B_{1i} \text{ y, } \dots, \hat{x}_n(k+1) \text{ es } B_{ni} \\ & \text{ o } \hat{x}_1(k+1) \text{ es } u_1 C_{1i} \text{ y, } \dots, \hat{x}_m(k+1) \text{ es } u_m C_{mi}, \dots, \hat{x}_n(k+1) \text{ es } C_{ni} \end{aligned} \quad (4.1)$$

entonces se puede escribir de las dos formas siguientes:

$$\begin{aligned} R_1^i : & \text{ si } \hat{x}_1(k) \text{ es } A_{1i} \text{ y } \hat{x}_2(k) \text{ es } A_{2i} \text{ y, } \dots, \hat{x}_n(k) \text{ es } A_{ni} \\ & \text{ entonces } \frac{\beta}{2} \hat{x}_1(k+1) \text{ es } B_{1i} \text{ y, } \dots, \frac{\beta}{2} \hat{x}_n(k+1) \text{ es } B_{ni} \\ R_2^i : & \text{ si } \hat{x}_1(k) \text{ es } A_{1i} \text{ y } \hat{x}_2(k) \text{ es } A_{2i} \text{ y, } \dots, \hat{x}_n(k) \text{ es } A_{ni} \\ & \text{ entonces } \frac{\beta}{2} \hat{x}_1(k+1) \text{ es } u_1 C_{1i} \text{ y, } \dots, \frac{\beta}{2} \hat{x}_m(k+1) \text{ es } u_m C_{mi}, \dots, \frac{\beta}{2} \hat{x}_n(k+1) \text{ es } C_{ni} \end{aligned} \quad (4.2)$$

donde  $\beta$  es conocida y es una constante positiva, se tiene que  $l$  ( $i = 1, 2, \dots, l$ ) la regla difusa (si entonces) realiza un mapeo de un vector de entrada  $X = [\hat{x}_1, \dots, \hat{x}_n] \in \mathbb{R}^n$  a un vector de

salida  $\hat{Y}(k) = [\hat{x}_1(k+1), \dots, \hat{x}_m(k+1)]^T$ .

$A_{1i}, \dots, A_{ni}, B_{1i}, \dots, B_{ni}$  y  $C_{1i}, \dots, C_{ni}$  son conjuntos difusos básicos. Para entrada variable  $x_i$  hay  $l_i$  conjuntos difusos. En el caso de conexión completa,  $l = l_1 \times l_2 \times \dots \times l_n$ , usando la consecuencia del producto, centro promedio y la salida  $pth$  del sistema de lógica difusa se expresa como.

$$\begin{aligned} \hat{x}_p(k+1) &= \frac{\beta}{2} \left( \sum_{i=1}^l w_{1pi} \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] \right) / \left( \sum_{i=1}^l \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] \right) = \frac{\beta}{2} \sum_{i=1}^l w_{1pi} \phi_i \\ \hat{x}_p(k+1) &= \frac{\beta}{2} \left( \sum_{i=1}^l u_i w_{2pi} \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] \right) / \left( \sum_{i=1}^l \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] \right) = \frac{\beta}{2} \sum_{i=1}^l u_i w_{2pi} \phi_i \end{aligned} \quad (4.3)$$

donde  $\mu_{A_{ji}}$  es una función pertenencia del conjunto difuso  $A_{ji}$ ,  $w_{pi}$  es el punto en el cual  $\mu_{B_{pi}} = 1$ . Si definimos

$$\phi_i = \prod_{j=1}^n \mu_{A_{ji}} / \sum_{i=1}^l \prod_{j=1}^n \mu_{A_{ji}} \quad (4.4)$$

la ecuación (4.3) se puede expresar en forma de matriz como

$$\beta \hat{x}(k+1) = W_1(k) \sigma[x(k)] + W_2(k) \sigma[x(k)] U(k) \quad (4.5)$$

donde los parámetros,

$$W_k = \begin{bmatrix} w_{11} & & w_{1l} \\ & \ddots & \\ w_{n1} & & w_{nl} \end{bmatrix} \quad (4.6)$$

el vector de datos  $\sigma[x(k)] = [\phi_1 \cdots \phi_l]^T$ ,  $U(k) = [u_1, u_2 \cdots u_m, 0, \dots, 0]^T$ . La estructura del sistema de la estática neuronal difuso se muestra en la Figura (3.1). Estas cuatro capas de la red neuronal difuso fue estudiado por [14]. La estructura del sistema dinámicos neuronal difuso se muestra en la Figura (4.1). La identificación del sistema no lineal se muestra como (2.17), asumimos que la planta es estable BIBO, y entonces  $x(k)$  y  $u(k)$  son acotados. Segun el teorema Stone Weierstrass, en general la función suave no lineal se escribe como:

$$\beta x(k+1) = W_1^* \sigma[x(k)] + W_2^* \sigma[x(k)] U(k) + \mu(k) \quad (4.7)$$

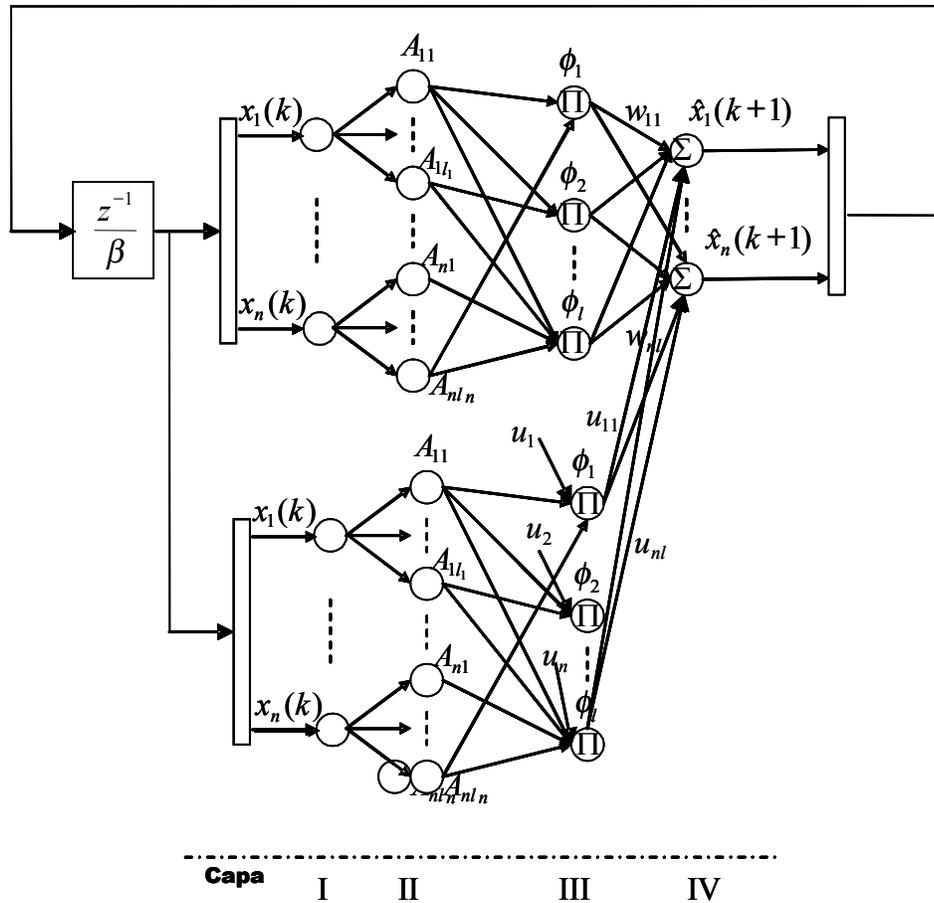


Figura 4.1: La estructura del sistema de la dinámica neuronal difuso

donde  $W_1^*$  y  $W_2^*$  son pesos constantes que pueden minimizar  $\mu(k)$ , el error a modelar es  $\mu(k)$

$$e(k) = \hat{x}(k) - x(k) \quad (4.8)$$

Ahora de la ecuación (4.7) tenemos

$$\beta e(k+1) = \widetilde{W}_1(k) \sigma[x(k)] + \widetilde{W}_2(k) \sigma[x(k)] U(k) + \zeta(k) \quad (4.9)$$

donde  $\widetilde{W}_1(k) = W_1(k) - W_1^*$ ,  $\widetilde{W}_2(k) = W_2(k) - W_2^*$ ,  $\varepsilon_1(k)$  el error es una aproximación de segundo orden,  $\sigma'$  es la derivada de la función de activación no lineal  $\sigma(\cdot)$  en el punto de  $W_1(k)$ . Después la función de activación es una sigmoide  $\phi$ ,  $\varepsilon_1(k)$  es acotada como  $\|\varepsilon_1(k)\|^2 \leq \bar{\varepsilon}_1$ ,  $\bar{\varepsilon}_1$  es una constante positiva no conocida. Donde  $\zeta(k) = \varepsilon_1(k) + \varepsilon_2(k) - \mu(k)$ . En el siguiente teorema tenemos un algoritmo de aprendizaje estable de tiempo discreto de una red neuronal de capa simple.

#### 4.1.2. Estructura tipo takagi sugeno dinámica difusa redes neuronales

$$R^i: \text{ Si } x_1 \text{ es } A_{1i} \text{ y } x_2 \text{ es } A_{2i} \text{ y } , \dots , x_n \text{ es } A_{ni} \text{ entonces } \hat{y}_j = p_{j0}^i + p_{j1}^i x_1 + \dots + p_{jn}^i x_n \quad (4.10)$$

donde  $j = 1, \dots, m$ . el  $p$ th salida del sistema de lógica difusa puede expresarse como

$$\hat{y}_p = \sum_{i=1}^l (p_{p0}^i + p_{p1}^i x_1 + \dots + p_{pn}^i x_n) \phi_i \quad (4.11)$$

donde  $\phi_i$  es definido como en la ecuación (4.4). Ahora la ecuación (4.11) puede ser expresado en la forma del tipo Mamdani ecuación (4.5),

$$\hat{Y}(k) = W(k) \sigma[x(k)]$$

donde  $\hat{Y}(k) = [\hat{y}_1 \dots \hat{y}_m]^T$

$$W(k) = \begin{bmatrix} p_{10}^1 \dots p_{10}^l & p_{11}^1 \dots p_{11}^l & \dots & p_{1n}^1 \dots p_{1n}^l \\ \vdots & \vdots & & \vdots \\ p_{m0}^1 \dots p_{m0}^l & p_{m1}^1 \dots p_{m1}^l & \dots & p_{mn}^1 \dots p_{mn}^l \end{bmatrix} \quad (4.12)$$

$$\sigma [x(k)] = \left[ \phi_1 \cdots \phi_l \quad x_1 \phi_1 \cdots x_1 \phi_l \quad \cdots \quad x_n \phi_1 \cdots x_n \phi_l \right]^T \quad (4.13)$$

## 4.2. Sistemas de identificación con dinámica neuro difuso

### 4.2.1. Funciones pertenencia conocidas

El interés principal es entender algunos conceptos de estabilidad de entrada estado (ISS). Consideramos el siguiente sistema no lineal de tiempo discreto.

$$x(k+1) = f[x(k), u(k)] \quad (4.14)$$

donde  $u(k) \in \mathbb{R}^m$  es el vector de entrada,  $x(k) \in \mathbb{R}^n$  un vector de estado, y  $y(k) \in \mathbb{R}^l$  es la salida del vector  $f$  es en general una función suave no lineales,  $f \in C^\infty$ . Ahora las definiciones (2.1, 2.2) y el teorema (2.3).

**Teorema 4.1** *Si se usa la red neuronal de capa simple para identificar la planta no lineal y  $A$  esta en un intervalo como  $-1 < \lambda(A) < 0$ , el siguiente actualización de la regla gradiente sin modificación robusta se puede identificar el error acotado  $e(k)$  (estable en  $L_\infty$ )*

$$\begin{aligned} W_1(k+1) &= W_1(k) - \eta(k) \sigma e(k)^T \\ W_2(k+1) &= W_2(k) - \eta(k) u(k) \sigma e^T(k) \end{aligned} \quad (4.15)$$

donde si

$$\beta \|e(k+1)\| \geq \|e(k)\|, \quad \eta(k) = \frac{\eta}{1 + \|\sigma\|^2 + \|u\sigma\|^2}$$

Si

$$\beta \|e(k+1)\| < \|e(k)\|, \quad \eta(k) = 0$$

$$0 < \eta \leq 1$$

**Demostración.** Seleccionamos la función de Lyapunov como

$$V(k) = \left\| \widetilde{W}_1(k) \right\|^2 + \left\| \widetilde{W}_2(k) \right\|^2 \quad (4.16)$$

donde  $\left\| \widetilde{W}_1(k) \right\|^2 = \sum_{i=1}^n \widetilde{w}_1(k)^2 = \text{tr} \left\{ \widetilde{W}_1^T(k) \widetilde{W}_1(k) \right\}$ . De la regla de actualización (5.23)

$$\widetilde{W}_1(k+1) = \widetilde{W}_1(k) - \eta(k) \sigma e(k)^T \quad (4.17)$$

Por eso,

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) \\ &= \left\| \widetilde{W}_1(k) - \eta(k) \sigma e(k)^T \right\|^2 - \left\| \widetilde{W}_1(k) \right\|^2 + \left\| \widetilde{W}_2(k) - \eta(k) u(k) \sigma e^T(k) \right\|^2 - \left\| \widetilde{W}_2(k) \right\|^2 \\ &= \eta^2(k) \|e(k)\|^2 \|\sigma\|^2 - 2\eta(k) \left\| \sigma \widetilde{W}_1(k) e^T(k) \right\| \\ &\quad + \eta^2(k) \|e(k)\|^2 \|u(k)\sigma\|^2 - 2\eta(k) \left\| u(k) \sigma \widetilde{W}_2(k) e^T(k) \right\| \end{aligned} \quad (4.18)$$

Usando la ecuación (4.9) y  $\eta(k) \geq 0$ ,  $\lambda_{\max}(A) \leq -\alpha$ ,  $\alpha > 0$ , ahí existe un pequeño  $\beta > 0$ , tal que  $\|\beta e(k+1)\| \geq \|e(k)\|$

$$\begin{aligned} & -2\eta(k) \left\| \sigma \widetilde{W}_1(k) e^T(k) \right\| - 2\eta(k) \left\| u(k) \sigma \widetilde{W}_2(k) e^T(k) \right\| \\ & \leq -2\eta(k) \|e^T(k)\| \|\beta e(k+1) - \zeta(k)\| \\ & = -2\eta(k) \|e^T(k) \beta e(k+1) - e^T(k) \zeta(k)\| \\ & \leq -2\eta(k) \|e^T(k) \beta e(k+1)\| + 2\eta(k) \|e^T(k) \zeta(k)\| \\ & \leq -2\eta(k) \|e(k)\|^2 + \eta(k) \|e(k)\| + \eta(k) \|\zeta(k)\|^2 \end{aligned} \quad (4.19)$$

después  $0 < \eta \leq 1$ , ,

$$\begin{aligned} \Delta V(k) &\leq \eta^2(k) \|e(k)\|^2 \|\sigma'x(k)\|^2 + \eta^2(k) \|e(k)\|^2 \left\| u(k) \phi'x(k) \right\|^2 \\ &\quad - \eta(k) \|e(k)\|^2 + \eta(k) \|\zeta(k)\|^2 \\ &= -\eta(k) \left[ 1 - \eta \frac{\|\sigma'x(k)\|^2 + \left\| u(k) \phi'x(k) \right\|^2}{1 + \|\sigma'x(k)\|^2 + \left\| u(k) \phi'x(k) \right\|^2} \right] e^2(k) + \eta \zeta^2(k) \\ &\leq -\pi e^2(k) + \eta \zeta^2(k) \end{aligned} \quad (4.20)$$

donde  $\pi = \frac{\eta}{1+\kappa} \left[ 1 - \frac{\kappa}{1+\kappa} \right]$ ,  $\kappa = \max_k \left( \|\sigma'x(k)\|^2 + \left\| u(k) \phi'x(k) \right\|^2 \right)$ . Puesto que  $-1 < \lambda(A) < 0$ ,  $\pi > 0$

$$n \min(\widetilde{w}_i^2) \leq V_k \leq n \max(\widetilde{w}_i^2) \quad (4.21)$$

donde  $n \times \min(\tilde{w}_i^2)$  y  $n \times \max(\tilde{w}_i^2)$  son funciones  $\mathcal{K}_\infty$ , y  $\pi e^2(k)$  es una función  $\mathcal{K}_\infty$ ,  $\eta \zeta^2(k)$  es una función  $\mathcal{K}$ , por eso  $V_k$  admite una función suave de Lyapunov ISS como en la *Definición (2.2)*. Del *teorema (2.3)*, la identificación del error dinámico es ISS. La "ENTRADA" corresponde al segundo término de la última línea en la ecuación (4.20), entonces, el error a modelar es  $\zeta(k) = \varepsilon_1(k) + \varepsilon_2(k) - \mu(k)$ , el "ESTADO" corresponde al primer término de la última línea de la ecuación (4.20), y entonces, la identificación del error es  $e(k)$  por que la "ENTRADA"  $\zeta(k)$  es acotada y la dinámica es ISS, el "ESTADO"  $e(k)$  es acotado. Cuando  $\beta \|e(k+1)\| < \|e(k)\|$ ,  $\Delta V(k) = 0$ .  $V(k)$  es constante, la constante de  $W_1(k)$  entonces  $e(k)$  es acotado (ver (4.9)). ■

**Propiedad 4.2** *La ecuación (4.15) es el algoritmo gradiente descendiente (BackPropagation), el cual el aprendizaje normalizado  $n_k$  es de tiempo variante en orden asegurar el proceso de identificación sea estable. Esta regla de aprendizaje es simple a usar, por que no se necesita tener cuidado como seleccionar un ley de aprendizaje mejor para asegurar ambos en la rapidez de convergencia y estabilidad. En ninguna información previa se requiere.*

**Propiedad 4.3** *La condición  $\beta \|e(k+1)\| \geq \|e(k)\|$  es la zona muerta. Si  $\beta$  es seleccionado muy grande, la zona muerta llega a ser pequeña.*

## 4.2.2. Funciones pertenencia desconocidas

Cuando la planta a identificar es considerada como caja negra, nosotros no podemos construir reglas difusas para la proposición parte de  $\mu_{A_{ji}}$  (ver [28], [30], [34]). El objeto del modelado de la neurona difusa es encontrar valores centrales ( $W_k$ ) de  $B_{1i} \cdots B_{mi}$ , como serían las funciones pertenencia  $A_{1i} \cdots A_{ni}$ , tal que la red neuronal difusa, puede seguir la planta no lineal.

La función pertenencia gaussiana para identificar las reglas difusas, son definidos como

$$\mu_{A_{ji}} = \exp\left(-\frac{(x_j - c_{ji})^2}{\rho_{ji}^2}\right) \quad (4.22)$$

$q$ th es la salida del modelo neuronal difuso se puede expresar de la forma

$$\hat{y}_q = \sum_{i=1}^l w_{qi} \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\rho_{ji}^2} \right) / \left[ \sum_{i=1}^l \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\rho_{ji}^2} \right) \right] \quad (4.23)$$

Ahora definimos

$$z_i = \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\rho_{ji}^2} \right), \quad a_q = \sum_{i=1}^l w_{qi} z_i, \quad b = \sum_{i=1}^l z_i \quad (4.24)$$

por eso

$$\hat{y}_q = \frac{a_q}{b} \quad (4.25)$$

Similar a la identificación del proceso no lineal, se puede expresar como

$$y_q = \sum_{i=1}^l w_{qi}^* \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji}^*)^2}{\rho_{ji}^{*2}} \right) / \left[ \sum_{i=1}^l \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji}^*)^2}{\rho_{ji}^{*2}} \right) \right] - \mu_q \quad (4.26)$$

donde  $w_{qi}^*$ ,  $c_{ji}^*$  y  $\rho_{ji}^{*2}$  son parámetros desconocidos el cual quizá no minimicen el modelado dinámico  $\mu_q$ .

En el caso de tres variables independientes, una función suave  $f$  tiene la formula de Taylor como:

$$f(x_1, x_2, x_3) = \sum_{k=0}^{l-1} \frac{1}{k!} \left[ (x_1 - x_1^0) \frac{\partial}{\partial x_1} + (x_2 - x_2^0) \frac{\partial}{\partial x_2} + (x_3 - x_3^0) \frac{\partial}{\partial x_3} \right]^k f + R_l \quad (4.27)$$

donde  $R_l$  es la formula de Taylor a recordar. Si  $x_1, x_2, x_3$  corresponde a  $w_{pi}^*$ ,  $c_{ji}^*$  y  $\sigma_{ji}^{*2}$ ,  $x_1^0, x_2^0, x_3^0$  corresponde a  $w_{pi}$ ,  $c_{ji}$  y  $\sigma_{ji}^2$ ,

$$y_q + \mu_q = \hat{y}_q + \sum_{i=1}^l (w_{qi}^* - w_{qi}) z_i / b + \sum_{i=1}^l \sum_{j=1}^n \frac{\partial}{\partial c_{ji}} \left( \frac{a_q}{b} \right) (c_{ji}^* - c_{ji}) \quad (4.28)$$

$$+ \sum_{i=1}^l \sum_{j=1}^n \frac{\partial}{\partial \rho_{ji}^2} \left( \frac{a_q}{b} \right) (\rho_{ji}^{*2} - \rho_{ji}^2) + R_{1q} \quad (4.29)$$

Usando la regla de la cadena, obtenemos lo siguiente

$$\begin{aligned} \frac{\partial}{\partial c_{ji}} \left( \frac{a_q}{b} \right) &= \frac{\partial}{\partial z_i} \left( \frac{a_q}{b} \right) \frac{\partial z_i}{\partial c_{ji}} = \left( \frac{1}{b} \frac{\partial a_q}{\partial z_i} + \frac{\partial}{\partial z_i} \left( \frac{1}{b} \right) a_q \right) \left( 2z_i \frac{x_j - c_{ji}}{\sigma_{ji}^2} \right) \\ &= \left( \frac{w_{qi}}{b} - \frac{a_q}{b^2} \right) \left( 2z_i \frac{x_j - c_{ji}}{\sigma_{ji}^2} \right) = 2z_i \frac{w_{qi} - \hat{y}_q}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} \end{aligned} \quad (4.30)$$

$$\frac{\partial}{\partial \rho_{ji}} \left( \frac{a_q}{b} \right) = \frac{\partial}{\partial z_i} \left( \frac{a_q}{b} \right) \frac{\partial z_i}{\partial \rho_{ji}} = 2z_i \frac{w_{qi} - \hat{y}_q}{b} \frac{(x_j - c_{ji})^2}{\rho_{ji}^3} \quad (4.31)$$

En forma de matriz,

$$y_q + \mu_q = \hat{y}_q - \widetilde{W}_q Z(k) - D_{Zq} \overline{C}_k E - D_{Zq} \overline{B}_k E \quad (4.32)$$

de donde

$$\begin{aligned} Z(k) &= [z_1/b \cdots z_l/b]^T, & W_q &= [w_{q1} \cdots w_{ql}], & \widetilde{W}_q &= W_q - W_q^* \\ D_{Zq} &= \left[ 2z_1 \frac{w_{q1} - \hat{y}_q}{b}, \dots, 2z_l \frac{w_{ql} - \hat{y}_q}{b} \right], & E &= [1, \dots, 1]^T \\ \overline{C}_k &= \begin{bmatrix} \frac{x_1 - c_{11}}{\sigma_{11}^2} (c_{11} - c_{11}^*) & & \frac{x_n - c_{n1}}{\sigma_{n1}^2} (c_{n1} - c_{n1}^*) \\ & \ddots & \\ \frac{x_1 - c_{1l}}{\sigma_{1l}^2} (c_{1l} - c_{1l}^*) & & \frac{x_n - c_{nl}}{\sigma_{nl}^2} (c_{nl} - c_{nl}^*) \end{bmatrix} \\ \overline{B}_k &= \begin{bmatrix} \frac{(x_1 - c_{11})^2}{\rho_{11}^3} (\rho_{11} - \rho_{11}^*) & & \frac{(x_n - c_{n1})^2}{\rho_{n1}^3} (\rho_{n1} - \rho_{n1}^*) \\ & \ddots & \\ \frac{(x_1 - c_{1l})^2}{\rho_{1l}^3} (\rho_{1l} - \rho_{1l}^*) & & \frac{(x_n - c_{nl})^2}{\rho_{nl}^3} (\rho_{nl} - \rho_{nl}^*) \end{bmatrix} \end{aligned} \quad (4.33)$$

definimos la identificación del error de la forma como se define

$$\begin{aligned} e_q &= \hat{y}_q - y_q \\ e_q &= Z(k) \widetilde{W}_q + D_{Zq} \overline{C}_k E + D_{Zq} \overline{B}_k E + \mu_q - R_{1q} \end{aligned} \quad (4.34)$$

donde  $R_{1q}$  es de segundo orden el error de aproximación, de la serie de Taylor,  $q = 1 \cdots m$ .

Nosotros definimos

$$\begin{aligned} e(k) &= [e_1 \cdots e_m]^T \\ \beta e(k) &= \widetilde{W}_k Z(k) + D_z(k) \overline{C}_k E + D_z(k) \overline{B}_k E + \zeta(k) \end{aligned} \quad (4.35)$$

$$\text{del cual } \widetilde{W}_k = \begin{bmatrix} w_{11} - w_{11}^* & & w_{m1} - w_{m1}^* \\ & \ddots & \\ w_{1l} - w_{1l}^* & & w_{ml} - w_{ml}^* \end{bmatrix}, \quad D_z(k) = \begin{bmatrix} 2z_1 \frac{w_{11} - \hat{y}_1}{b} & & 2z_l \frac{w_{1l} - \hat{y}_1}{b} \\ & \ddots & \\ 2z_1 \frac{w_{m1} - \hat{y}_m}{b} & & 2z_l \frac{w_{ml} - \hat{y}_m}{b} \end{bmatrix}.$$

$\zeta(k) = \mu - R_1$ ,  $\mu = [\mu_1 \cdots \mu_m]^T$ ,  $R_1 = [R_{11} \cdots R_{1m}]^T$ . Para las cotas de la función gaussiana  $\phi$  quizá asumimos que  $\mu$  en la ecuación (4.26) es acotado, también  $R_1$  es acotado. Por eso  $\zeta(k)$  en la ecuación (4.35) es acotado. El siguiente teorema nos da un algoritmo de retroalimentación estable para una red neuronal difusa en tiempo discreto.

**Teorema 4.4** *Si usamos la red neuronal difusa ecuación (4.23) para identificar la planta no lineal, el siguiente algoritmo de retroalimentación puede hacer la identificación del error  $e(k)$  acotado*

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) Z(k)^T \\ c_{ji}(k+1) &= c_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{x_j - c_{ji}}{\rho_{ji}^2} (\hat{y}_q - y_q) \\ \rho_{ji}(k+1) &= \rho_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{(x_j - c_{ji})^2}{\rho_{ji}^3} (\hat{y}_q - y_q) \end{aligned} \quad (4.36)$$

donde si

$$\beta \|e(k+1)\| \geq \|e(k)\|, \quad \eta_k = \frac{\eta}{1 + \|Z\|^2 + 2\|D_z\|^2}$$

Si

$$\beta \|e(k+1)\| < \|e(k)\|, \quad \eta_k = 0$$

$0 < \eta \leq 1$ . El error promedio de la identificación satisface

$$J = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T e^2(k) \leq \frac{\eta \bar{\zeta}}{\pi} \quad (4.37)$$

$$\text{donde } \pi = \frac{\eta}{1 + \kappa} \left[ 1 - \frac{\kappa}{1 + \kappa} \right] > 0, \quad \kappa = \max_k (\|Z\|^2 + 2\|D_z\|^2), \quad \bar{\zeta} = \max_k [\zeta^2(k)]$$

**Demostración.** Definimos  $\tilde{c}_{ji}(k) = c_{ji}(k) - c_{ji}^*$ ,  $\tilde{b}_{ji}(k) = \sigma_{ji}(k) - \sigma^*$ , el elemento de  $\tilde{C}_k$  es expresado como  $\tilde{c}_{ji}(k) = [\tilde{C}_k]$ . Por eso

$$[\tilde{C}_{k+1}] = [\tilde{C}_k] - 2\eta_k z_i \frac{w_{qi} - \hat{y}_q}{b} \frac{x_j - c_{ji}}{\rho_{ji}^2} (\hat{y}_q - y_q) \quad (4.38)$$

Nosotros seleccionamos una matriz definida positiva  $L_k$  como

$$L_k = \|\tilde{W}_k\|^2 + \|\tilde{C}_k\|^2 + \|\tilde{B}_k\|^2 \quad (4.39)$$

Usando la ecuación (4.35) tenemos

$$\begin{aligned}
\Delta L_k &= \left\| \widetilde{W}_k - \eta_k e(k) Z(k)^T \right\|^2 + \left\| \widetilde{C}_k - \left[ 2\eta_k z_i \frac{w_{qi} - \hat{y}_q}{b} \frac{x_j - c_{ji}}{\rho_{ji}^2} (\hat{y}_q - y_q) \right] \right\|^2 \\
&+ \left\| \widetilde{B}_k - \left[ 2\eta_k z_i \frac{w_{qi} - \hat{y}_q}{b} \frac{(x_j - c_{ji})^2}{\rho_{ji}^3} (\hat{y}_q - y_q) \right] \right\|^2 - \left\| \widetilde{W}_k \right\|^2 - \left\| \widetilde{C}_k \right\|^2 - \left\| \widetilde{B}_k \right\|^2 \\
&= \eta_k^2 e^2(k) \left( \|Z(k)^T\|^2 + 2\|D_z^T\|^2 \right) - 2\eta_k \|e(k)\| \left\| \widetilde{W}_k Z(k)^T + D_z^T \widetilde{C}_k E + D_z^T \widetilde{B}_k E \right\| \\
&= \eta_k^2 e^2(k) (\|Z\|^2 + 2\|D_z\|^2) - 2\eta_k \|e(k)\| [e(k) - \zeta(k)] \\
&\leq -\eta_k e^2(k) [1 - \eta_k (\|Z\|^2 + 2\|D_z\|^2)] + \eta \zeta^2(k) \\
&\leq -\pi e^2(k) + \eta \zeta^2(k)
\end{aligned} \tag{4.40}$$

donde  $\pi$  se define como

$$\pi = \frac{\eta}{1 + \max_k (\|Z\|^2 + 2\|D_z\|^2)} \tag{4.41}$$

Por que

$$n \left[ \min(\tilde{w}_i^2) + \min(\tilde{c}_{ji}^2) + \min(\tilde{b}_{ji}^2) \right] \leq L_k \leq n \left[ \max(\tilde{w}_i^2) + \max(\tilde{c}_{ji}^2) + \max(\tilde{b}_{ji}^2) \right] \tag{4.42}$$

donde  $n \left[ \min(\tilde{w}_i^2) + \min(\tilde{c}_{ji}^2) + \min(\tilde{b}_{ji}^2) \right]$  y  $n \left[ \max(\tilde{w}_i^2) + \max(\tilde{c}_{ji}^2) + \max(\tilde{b}_{ji}^2) \right]$  son funciones  $\mathcal{K}_\infty$ , y  $\pi e^2(k)$  es una función  $\mathcal{K}_\infty$ ,  $\eta \zeta^2(k)$  es una función  $\mathcal{K}$ . De las ecuaciones (4.35) y (4.39) conocemos que  $V_k$  es la función de  $e(k)$  y  $\zeta(k)$ , por eso  $L_k$  admite una función de Lyapunov suave ISS como en la *Definición (2.2)*. del *Teorema (2.3)*, la dinámica del error de identificación entrada a estado estable ISS. Por que la "ENTRADA"  $\zeta(k)$  Es acotada y la dinámica es ISS, el "ESTADO"  $e(k)$  es acotado.

La ecuación (5.46) se puede reescribir como

$$\Delta L_k \leq -\pi e^2(k) + \eta \zeta^2(k) \leq \pi e^2(k) + \eta \bar{\zeta} \tag{4.43}$$

Sumando la ecuación (4.43) de 1 hasta  $T$ , y usando  $L_T > 0$  y  $L_1$  es una constante, obtenemos

$$\begin{aligned}
L_T - L_1 &\leq -\pi \sum_{k=1}^T e^2(k) + T\eta \bar{\zeta} \\
\pi \sum_{k=1}^T e^2(k) &\leq L_1 - L_T + T\eta \bar{\zeta} \leq L_1 + T\eta \bar{\zeta}
\end{aligned} \tag{4.44}$$

la ecuación (3.22) es establecido. ■

**Propiedad 4.5** Para el modelo neuronal difuso del tipo Takagi Sugeno Kang ecuación (4.10) con una función gaussiana pertenencia en  $A_{ji}$ , el  $q$ th salida del sistema lógico difuso puede expresarse como

$$\hat{y}_q = \sum_{i=1}^l \left( \sum_{k=0}^n p_{qk}^i x_k \right) \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) / \left[ \sum_{i=1}^l \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) \right] \quad (4.45)$$

donde  $x_0 = 1$ . La parte  $\sum_{i=1}^l (w_{qi}^* - w_{qi}) z_i / b_q$  es la ecuación (4.28) es cambiando como

$$\sum_{i=1}^l \left( \sum_{k=0}^n (p_{qk}^{i*} - p_{qk}^i) x_k \right) z_i / b \quad (4.46)$$

Si usamos el siguiente algoritmo

$$\begin{aligned} p_{qk}^i(k+1) &= p_{qk}^i(k) - \eta_k (\hat{y}_q - y_q) \frac{z_i}{b} x_k \\ c_{ji}(k+1) &= c_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{x_j - c_{ji}}{\sigma_{ji}^2} (\hat{y}_q - y_q) \\ \sigma_{ji}(k+1) &= \sigma_{ji}(k) - 2\eta_k z_i \frac{w_{pi} - \hat{y}_p}{b} \frac{(x_j - c_{ji})^2}{\sigma_{ji}^3} (\hat{y}_q - y_q) \end{aligned} \quad (4.47)$$

El error de identificación  $e(k)$  es acotado. La demostración es la misma como en el teorema (3.1).

**Propiedad 4.6** Normalizando el porcentaje de aprendizaje  $\eta_k$  en las ecuaciones (3.17) y (3.21) son de tiempo variante, el método del error de identificación. Estas ganancias de aprendizaje son fáciles para ser decididos como en [30], en ninguna información a priori se requiere, por ejemplo si nosotros quizá seleccionamos  $\eta = 1$ . La contradicción en la rápida convergencia y el algoritmo estable puede ser evitado. Cambiando el porcentaje del tiempo variante puede ser encontrando algunos esquemas adaptables básicos como en [21]. Pero ellos usan también modificación robusta para garantizar el proceso de identificación estable.

**Propiedad 4.7** Si seleccionamos  $\eta$  como una función de zona muerta:

$$\begin{cases} \eta = 0 & \text{si } |e(k)| \leq \bar{\zeta} \\ \eta = \eta_0 & \text{si } |e(k)| > \bar{\zeta} \end{cases} \quad (4.48)$$

la ecuación (3.21) es el mismo como en [35]. Si un término modificación  $\sigma$  o modificando la regla  $\delta$  los términos son sumados en  $k$  en la ecuación (3.17) o (3.21), Esto nos dice que de [34] o [26]. Pero todas ellas necesitan la mayor de la cota del modelado del error  $\bar{\zeta}$ . Y la identificación del error aumenta por las modificaciones de robustez [21].

**Propiedad 4.8** Puesto que asumimos que la red neuronal difusa no puede igualar sistemas no lineales, no podemos hacer que los parámetros ( $w_{ji}$ ,  $c_{ji}$  y  $\sigma_{ji}$ ) convergen, nos gustaría solo forzar la salida de la red neuronal difusa siguiendo a la salida de la planta, entonces la identificación del error sera estable. Sin embargo los pesos no pueden converger a sus valores óptimos, la ecuación (3.22) muestra que el error de identificación convergerá en una bola de radio  $\frac{\eta}{\pi}\bar{\zeta}$ . Si la entrada es constante, el modelado del error  $\zeta(k)$  no harán que los pesos convergen a sus valores óptimos. Esto hace posible que la salida del error converge, pero los pesos del error son muy altos cuando las reglas difusas no son bien definidas. La relación de la salida del error y el error de los pesos se muestran en las ecuaciones (??) y (4.35). En un caso simple usamos la ecuación (??) y las redes neuronales difusas no pueden igualar exactamente a la planta no lineal.

$$\begin{aligned}
 \text{planta: } y &= W^* \Phi [X(k)] \\
 \text{redes neuronales difusas: } \hat{y} &= W(k) \Phi [X(k)] \\
 \text{salida del error: } (y - \hat{y}) &= (W^* - W(k)) \Phi [X(k)]
 \end{aligned}
 \tag{4.49}$$

Si  $\Phi [X(k)]$  es grande, la salida del error es pequeño ( $y - \hat{y}$ ) no significa una buena convergencia del error de los pesos  $W^* - W(k)$ .

### 4.3. Simulación

1). Un sistema de tipo caótico, del modelo de Lorenz se elige para demostrar la facilidad del neuro identificador. El modelo de Lorenz se usa para el fluido de la descripción específicamente para algunos características de la dinámica atmosférica [41]. El modelo a controlar

es descrito de la forma:

$$\begin{aligned}\dot{x}_1 &= -\beta x_1 + x_2 x_3 \\ \dot{x}_2 &= \omega(x_3 - x_2) \\ \dot{x}_3 &= -x_1 x_2 + \rho x_2 - x_3\end{aligned}\tag{4.50}$$

donde  $x_1$ ,  $x_2$  y  $x_3$  representan medidas de temperaturas verticales, horizontales, velocidad de fluido correspondientes,  $\omega$ ,  $\rho$  y  $\beta$  son parámetros positivos representando el número de *Prandtl*, número *Rayleigh* y factor *geométrico*. Si  $\rho < 1$ , es el origen de equilibrio estable. Si

$$1 < \rho \leq \rho^*(\omega, \beta) := \omega(\omega + \beta + 3) / (\omega - \beta - 1),\tag{4.51}$$

el sistema tiene dos puntos de equilibrio  $(\pm\sqrt{\beta(\rho - 1)}, \pm\sqrt{\beta(\rho - 1)}, (\rho - 1))$ , y un punto de equilibrio inestable el cual es (el origen). Si  $\rho^*(\omega, \beta) < \rho$ , los tres puntos de equilibrio llegan a ser inestables. Como el caso comúnmente estudiado, seleccionamos  $\omega = 10$  y  $\beta = 8/3$  que lleva a  $\rho^*(\omega, \beta) = 24.74$ . En este ejemplo se considera el sistema con  $\omega = 10$ ,  $\beta = 8/3$  y  $\rho = 28$ . El sistema caótico dinámico se muestra en las Figuras (4.2, 4.3, 4.4).

Usamos la siguiente técnica de diferencia para obtener los estados de tiempo discreto del sistema ecuación (4.50), y de la ecuación (4.50) el cual se puede escribir de la forma siguiente

$$\dot{x} = Ax\tag{4.52}$$

con  $x = [x_1, x_2, x_3]^T$ ,  $A = \begin{bmatrix} -\beta & 0 & x_2 \\ 0 & -\sigma & \sigma \\ -x_2 & \rho & -1 \end{bmatrix}$ . Ahora definimos  $s_1 = Ax_k$ ,  $s_2 = A(x_k + s_1)$ ,

$s_3 = A(x_k + \frac{s_1 + s_2}{4})$ . Si  $|\frac{s_1 - 2*s_3 + s_2}{3}| \leq \frac{|x_k|}{1000}$  o  $|\frac{s_1 - 2*s_3 + s_2}{3}| < 1$ , entonces  $x_{k+1} = x_k + \frac{s_1 + 4s_3 + s_2}{6}$ ,  $k = 0, 1, 2 \dots$ ,  $x_0 = [50, 10, 20]^T$ .

2). Primero usaremos una red neuronal difusa recurrente con las reglas siguientes entonces se puede escribir de las dos formas siguientes

$R_1^i$  : si  $\hat{x}_1(k)$  es  $A_{1i}$  y  $\hat{x}_2(k)$  es  $A_{2i}$  y, ...,  $\hat{x}_n(k)$  es  $A_{ni}$

entonces  $\frac{\beta}{2}\hat{x}_1(k+1)$  es  $B_{1i}$  y, ...,  $\frac{\beta}{2}\hat{x}_n(k+1)$  es  $B_{ni}$

$R_2^i$  : si  $\hat{x}_1(k)$  es  $A_{1i}$  y  $\hat{x}_2(k)$  es  $A_{2i}$  y, ...,  $\hat{x}_n(k)$  es  $A_{ni}$

entonces  $\frac{\beta}{2}\hat{x}_1(k+1)$  es  $u_1 C_{1i}$  y, ...,  $\frac{\beta}{2}\hat{x}_m(k+1)$  es  $u_m C_{mi}$ , ...,  $\frac{\beta}{2}\hat{x}_n(k+1)$  es  $C_{ni}$

donde la función pertenencia se conoce como una función gaussiana

$$\phi_i = \prod_{j=1}^n \mu_{A_{ji}} / \sum_{i=1}^l \prod_{j=1}^n \mu_{A_{ji}} \quad (4.53)$$

de simple capa ecuación (4.5) para identificar la ecuación (4.50),

$$\beta \hat{x}(k+1) = A \hat{x}(k) + W_1(k) \sigma[x(k)] \quad (4.54)$$

de donde  $\sigma[x(k)] = [\phi_1 \cdots \phi_l]^T$ ,  $\beta = 4$ ,  $\sigma(\cdot) = \tanh(\cdot)$ ,  $\hat{x} = [\hat{x}_1, \hat{x}_2, \hat{x}_3]^T$ ,  $A = \text{diag}[0.8, 0.8, 0.8]$ ,  $W_1(k) \in \mathfrak{R}^{3 \times 3}$ . Los elementos de  $W_1(0)$  son números aleatorios entre  $[0, 1]$ . Usamos la ecuación (5.23) para actualizar los pesos para la identificación en línea.

$$W_1(k+1) = W_1(k) - \frac{1}{1 + \|\sigma\|^2} e(k) \sigma^T \quad (4.55)$$

Del teorema (4.1) tenemos una condición necesaria para un aprendizaje estable de  $\eta$ , con  $\eta \leq 1$ . En este ejemplo, encontramos que si  $\eta \geq 2.5$ , el proceso de aprendizaje llega a ser inestable. El neuro identificación en esta simulación esta en línea, aquí no se estudia la convergencia de los pesos, cuidamos el error de identificación  $e(k)$ . Los pesos no convergen a algunas constantes o valores óptimos. La simulación total en tiempo es de 600, ahí hay 14 tiempos  $\beta \|e(k+1)\| < e(k)$ . Usamos una red difusa neuronal multicapa para identificar (4.50),

$$\beta \hat{x}(k+1) = A \hat{x}(k) + V_{k\sigma} [W_k(K) x(k)] \quad (4.56)$$

donde  $\beta = 4$ ,  $\sigma(\cdot) = \tanh(\cdot)$ ,  $A = \text{diag}[\cdot 8, \cdot 8, \cdot 8]$ . La complejidad del modelo es importante en el contexto de la identificación del sistema, el cual corresponde al número de unidades ocultas del neuro modelo. En esta simulación probamos diferentes números de nodos ocultos, encontramos cuando los nodos ocultos son mas de 20, la exactitud de la identificación todavía no ha sido mejorado mucho. En [27], también usaron 20 neuronas ocultas para la primer capa oculta

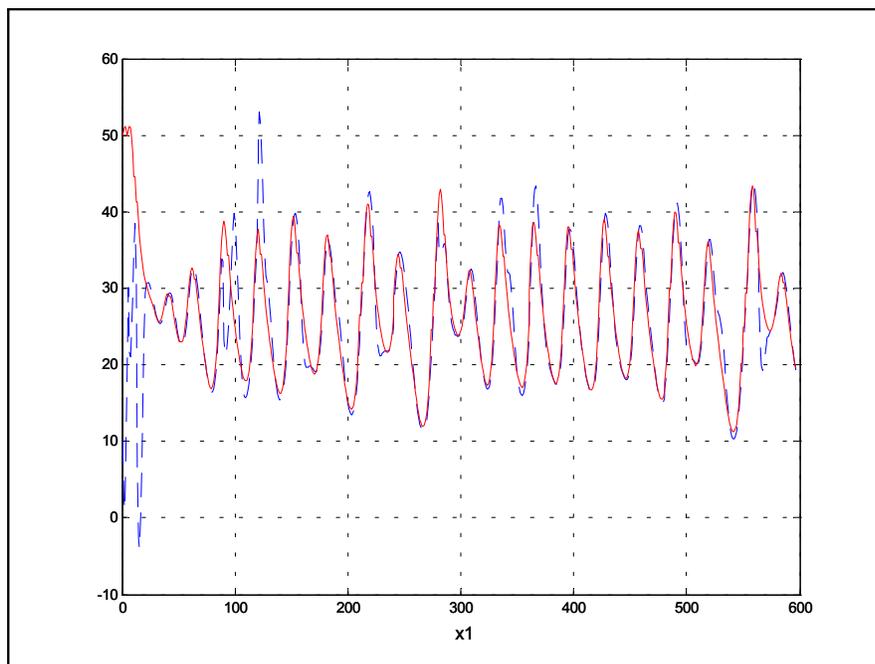


Figura 4.2: Sistema dinámico caótico de Lorenz

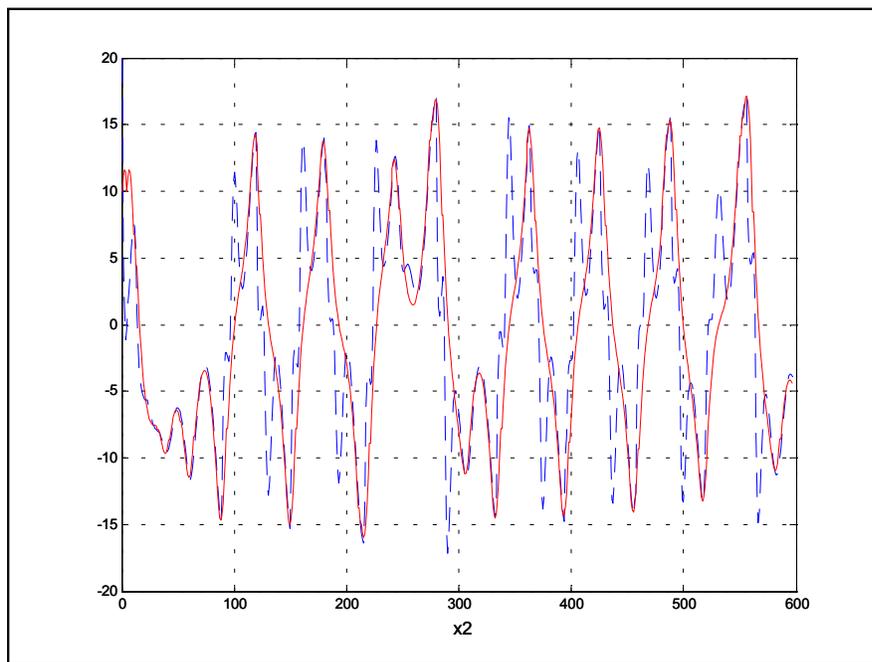


Figura 4.3: Sistema dinámico caótico de Lorenz

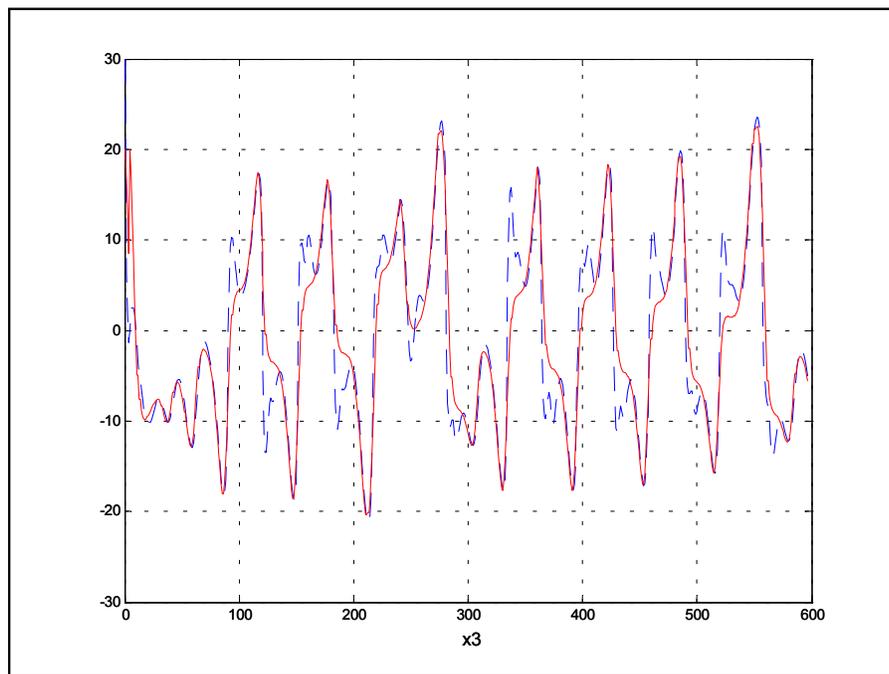


Figura 4.4: Sistema dinámico caótico de Lorenz

# Capítulo 5

## Redes neuronales difusas dinámicas para sistemas no lineales de control adaptable

### 5.1. Estructura difusa neuro control

El grado relativo del sistema, para entrada y salida simple,

$$y(k) = f[y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m)] + g[(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m)]u(k) \quad (5.1)$$

donde  $f[\cdot]$  y  $g[\cdot]$  son funciones suaves,  $g[\cdot]$  es acotada fuera del cero. Este sistema de entrada salida puede convertirse en una forma de espacios de estados, nosotros definimos

$$\begin{aligned} x_i(k) &= y(k-n+i-1), \quad i = 1 \dots n \\ x_{i+n}(k) &= u(k-m+i-1), \quad i = 1 \dots m \end{aligned} \quad (5.2)$$

Un modelo de estado de la ecuación (5.1) es

$$\begin{aligned} x_i(k+1) &= x_{i+1}(k), \quad i = 1 \dots n-1 \\ x_n(k+1) &= f[x(k)] + g[x(k)]u(k) \end{aligned} \quad (5.3)$$

donde  $x(k) = [x_1 \cdots x_{n+m}]^T$ ,  $g[x(k)]$  es diferente de cero con la señal conocida. Entonces, sin perder la generalidad,  $g[x(k)] \geq \bar{g} > 0$ ,  $\bar{g}$  es una constante positiva conocida. Dando un diseño de la trayectoria  $x_n^*(k)$  y estos valores de retardo  $x_i^*(k+1) = x_{i+1}^*(k)$ , el error acumulado se define como

$$e_{n-i}(k) = x_{n-i}(k) - x_{n-i}^*(k), \quad i = 0 \cdots n-1 \quad (5.4)$$

Un filtro de seguimiento de error es usado típicamente en un controlador de robot como sigue [26]

$$r(k) = e_n(k) + \lambda_1 e_{n-1}(k) + \cdots + \lambda_{n-1} e_1(k) \quad (5.5)$$

donde  $\lambda_1 \cdots \lambda_{n-1}$  son constantes seleccionados para que  $(z^{n-1} + \lambda_1 z^{n-2} + \cdots + \lambda_{n-1})$  sea estable. Usando las ecuaciones (5.3), (5.5) y  $r(k+1) = e_n(k+1) + \lambda_1 e_{n-1}(k+1) + \cdots + \lambda_{n-1} e_1(k+1)$ , la dinámica del error de seguimiento es

$$r(k+1) = f[x(k)] + g[x(k)]u(k) - x_n^*(k+1) + \lambda_1 e_n(k) + \cdots + \lambda_{n-1} e_2(k) \quad (5.6)$$

Podemos seleccionar el estado de control de retroalimentación como

$$u(k) = \frac{1}{g[x(k)]} [x_n^*(k+1) - f[x(k)] + k_v r(k) - \lambda_1 e_n(k) - \cdots - \lambda_{n-1} e_2(k)] \quad (5.7)$$

donde  $1 > |k_v|$ . El sistema de lazo cerrado llega a ser

$$r(k+1) = k_v r(k) \quad (5.8)$$

Por eso  $r(k)$  es asintóticamente estable. Puesto que la ecuación (5.5) se puede reescribir como

$$E(k+1) = AE(k) + Br(k) \quad (5.9)$$

donde  $E(k) = [e_1(k) \cdots e_{n-1}(k)]$ ,  $A = \begin{bmatrix} 0 & 1 & 0 \\ -\lambda_{n-1} & \cdots & -\lambda_1 \end{bmatrix}$ ,  $B = [0 \cdots 01]^T$ .

Este es un sistema lineal estable con entrada  $r(k)$ , por eso  $e_n(k)$  es asintóticamente estable.

**Propiedad 5.1** *El sistema de entrada y salida simple (SISO) el grado relativo de un sistema no lineal ecuación (5.1) en el caso simple, para sistemas de multi entradas y multi salidas (MIMO), la forma general del control de retroalimentación es similar (ver [22]). Si el grado relativo está más de uno, la transformación está más completa, pero el control de retroalimentación es similar como la ecuación (5.7), para más discusión puede encontrarse en [3].*

Si  $f[x(k)]$  y  $g[x(k)]$  en la ecuación (5.3) son redes neuronales conocidas en tiempo discreto se pueden aplicar para aproximar estas funciones. Recordando el teorema Stone Weierstrass [4], estas funciones suaves generales no lineales se pueden expresar por redes neuronales difusa de donde se puede representar como

Un modelo genérico difuso se presenta como una colección de reglas difusas en la forma siguiente:

$$R^i : \text{si } x_1 \text{ es } A_{1i} \text{ y } x_2 \text{ es } A_{2i} \text{ y, } \dots, x_n \text{ es } A_{ni} \text{ entonces } \hat{f}_1 \text{ es } B_{1i} \text{ y, } \dots, \hat{f}_m \text{ es } B_{mi} \quad (5.10)$$

Usamos  $l(i = 1, 2, \dots, l)$  difuso si entonces las reglas para realizar un vector de un mapeo de entrada lingüística  $X = [x_1, \dots, x_n] \in \mathbb{R}^n$  a una variable de salida lingüística.  $A_{1i}, \dots, B_{mi}$  son conjuntos difusos estandares [33]. Para la entrada  $x_i$  variables hay  $l_i$  conjunto difusos. En el caso de conexión completa,  $l = l_1 \times l_2 \times \dots, l_n$  de [33] sabemos. Usando la deducción del producto, centro promedio y semifallo del borrosificador, el  $k$ th salida del sistema lógica difusa se expresa como:

$$\begin{aligned} \hat{f}[x(k)]_p &= \left( \sum_{i=1}^l w_{pi} \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] / \sum_{i=1}^l \left[ \prod_{j=1}^n \mu_{A_{ji}} \right] \right) \\ &= \sum_{i=1}^l w_{pi} \phi_i \end{aligned} \quad (5.11)$$

donde  $\mu_{A_{ji}}$ , son las funciones de pertenencia del conjunto difuso  $A_{ji}$  y  $w_{pi}$  es el punto en el cual  $\mu_{B_{pi}} = 1$ . Si definimos

$$\phi_i = \prod_{j=1}^n \mu_{A_{ji}} / \sum_{i=1}^l \prod_{j=1}^n \mu_{A_{ji}}. \quad (5.12)$$

La ecuación (3.5) puede expresarse en parámetros

$$\widehat{f}[x(k)] = W_1 \phi_1[x(k)] \quad (5.13)$$

donde

$$\widehat{f}[x(k)] = [\widehat{f}_1, \dots, \widehat{f}_m]^T = W_1 = \begin{bmatrix} w_{11} & & w_{1l} \\ & \ddots & \\ w_{m1} & & w_{ml} \end{bmatrix} \quad (5.14)$$

el vector de datos  $\phi_1[x(k)] = [\phi_1, \dots, \phi_l]^T$ .

$$\begin{aligned} f[x(k)] &= W_1^* \phi_1[x(k)] - \widetilde{f}_1 \\ g[x(k)]u(k) &= W_2^* \phi_2[x(k)]u(k) - \widetilde{g}_1 \end{aligned} \quad (5.15)$$

donde  $\widetilde{f}_1$  y  $\widetilde{g}_1$  son funciones del error de evaluación  $W_i^* \in R^{1 \times n}$  es una matriz constante que puede minimizar el error de la función de aproximación. La función de vector de dimensión  $\phi_i$  y  $m$ , con  $i = 1, 2$ . La representación típica del elemento  $\phi_i(\cdot)$  es la función de activación sigmoide. Sea  $W_{1,k}$  y  $W_{2,k}$  denotando la estimación de  $W_1^*$  y  $W_2^*$ .

$$\begin{aligned} W_1^* \phi_1[x(k)] &= W_{1,k} \phi_1[x(k)] - \widetilde{f}_2, \\ W_2^* \phi_2[x(k)]u(k) &= W_{2,k} \phi_2[x(k)]u(k) - \widetilde{g}_2 \\ f[x(k)] &= \widehat{f}[x(k)] - \widetilde{f}, \quad g[x(k)]u(k) = \widehat{g}[x(k)]u(k) - \widetilde{g} \end{aligned} \quad (5.16)$$

donde  $\widetilde{f} = \widetilde{f}_1 + \widetilde{f}_2$  y  $\widetilde{g} = \widetilde{g}_1 + \widetilde{g}_2$  son errores de la función de estimación.

$$\begin{aligned} \widehat{f}[x(k)] &= W_{1,k} \phi_1[x(k)] \\ \widehat{g}[x(k)] &= W_{2,k} \phi_2[x(k)] \end{aligned} \quad (5.17)$$

La ecuación de redes neuronales difusas se puede expresar en forma de matriz como

$$\beta \widehat{x}(k+1) = W_1(k) \sigma[x(k)] + W_2(k) \sigma[x(k)]u(k) \quad (5.18)$$

## 5.2. Difusa neuro control con algoritmo estable

### 5.2.1. Aprendizaje sin funciones pertenencia

La dinámica del error de seguimiento se escribe

$$r(k+1) = W_{1,k}\phi_1[x(k)] + W_{2,k}\phi_2[x(k)]u(k) - x_n^*(k+1) + \lambda_1 e_n(k) + \dots + \lambda_{n-1}e_2(k) + \omega(k) \quad (5.19)$$

donde  $\omega(k) = \tilde{f} + \tilde{g}$ . Entonces el control  $u(k)$  se puede definir como lo siguiente

$$u(k) = [W_{2,k}\phi_2]^+ [x_n^*(k+1) - W_{1,k}\phi_1[x(k)] + k_v r(k) - \lambda_1 e_n(k) - \dots - \lambda_{n-1}e_2(k)] \quad (5.20)$$

donde  $[\cdot]^+$  puesto que el sentido de la pseudo inversa en Moor Penrose, que se define como [1].

$$x^+ = \frac{x^T}{\|x\|^2}, \quad 0^+ = 0 \quad (5.21)$$

El sistema de lazo cerrado es

$$\begin{aligned} r(k+1) &= k_v r(k) + \omega(k) \\ &= k_v r(k) + \tilde{f} + \tilde{g}u(k) \\ &= k_v r(k) + \widetilde{W}_{1,k}\phi_1[x(k)] + \widetilde{W}_{2,k}\phi_2[x(k)]u(k) + \omega_1(k) \end{aligned} \quad (5.22)$$

donde  $\widetilde{W}_{1,k} = W_{1,k} - W_{1,k}^*$ ,  $\widetilde{W}_{2,k} = W_{2,k} - W_{2,k}^*$ ,  $\omega_1(k) = \tilde{f}_1 + \tilde{g}_1$ . Recordando [20],  $\omega(k)$  y  $\omega_1(k)$  las constantes son acotadas  $\bar{\omega}$  y  $\bar{\omega}_1$ , entonces,  $\|\omega(k)\|^2 \leq \bar{\omega}$ ,  $\|\omega_1(k)\|^2 \leq \bar{\omega}_1$ .

**Teorema 5.2** *Si usamos el neuro control adaptable ecuación (5.20) para el control de la planta no lineal ecuación (5.3), la siguiente ley de actualización del gradiente sin la modificación robusta se puede hacer que el error de seguimiento sea acotado  $r(k)$  (estable en el sentido de  $L_\infty$ )*

$$\begin{aligned} W_{1,k+1} &= W_{1,k} - \eta_k r(k) \phi_1^T[x(k)] \\ W_{2,k+1} &= W_{2,k} - \eta_k r(k) \phi_2^T[x(k)]u(k) \end{aligned} \quad (5.23)$$

donde  $\eta_k$  satisface que

$$\eta_k = \begin{cases} \frac{\eta}{1 + \|\phi_1\|^2 + \|\phi_2 u\|^2} & \text{si } \beta \|r(k+1)\| \geq \|r(k)\| \\ 0 & \text{si } \beta \|r(k+1)\| < \|r(k)\| \end{cases} \quad (5.24)$$

aquí  $1 \geq \eta > 0$ ,  $\frac{1}{1+k_v} \geq \beta \geq 1$ .

**Demostración.** Seleccionamos una función de Lyapunov como

$$L_k = \left\| \widetilde{W}_{1,k} \right\|^2 + \left\| \widetilde{W}_{2,k} \right\|^2 \quad (5.25)$$

donde  $\left\| \widetilde{W}_{1,k} \right\|^2 = \sum_{i=1}^n \widetilde{w}_{1,k,i}^2 = \text{tr} \left\{ \widetilde{W}_{1,k}^T \widetilde{W}_{1,k} \right\}$ . De la ley de actualización ecuación (5.23)

$$\begin{aligned} \widetilde{W}_{1,k+1} &= \widetilde{W}_{1,k} - \eta_k r(k) \phi^T [x(k)] \\ \Delta L_k &= L_{k+1} - L_k \\ &= \left\| \widetilde{W}_{1,k} - \eta_k r(k) \phi_1^T [x(k)] \right\|^2 - \left\| \widetilde{W}_{1,k} \right\|^2 + \left\| \widetilde{W}_{2,k} - \eta_k r(k) \phi_2^T [x(k)] u(k) \right\|^2 - \left\| \widetilde{W}_{2,k} \right\|^2 \\ &= \eta_k^2 r^2(k) \left\| \phi_1 \right\|^2 + \eta_k^2 r^2(k) \left\| \phi_2 u \right\|^2 - 2\eta_k \left\| r(k) \phi_1^T \widetilde{W}_{1,k} \right\| - 2\eta_k \left\| r(k) \phi_2^T \widetilde{W}_{2,k} u(k) \right\| \end{aligned} \quad (5.26)$$

Usando la ecuación (5.22)

$$\begin{aligned} &-2\eta_k \left\| r(k) \phi_1^T \widetilde{W}_{1,k} \right\| - 2\eta_k \left\| r(k) \phi_2^T \widetilde{W}_{2,k} u(k) \right\| \\ &= -2\eta_k \|r(k)\| \left( \left\| \phi_1^T \widetilde{W}_{1,k} \right\| + \left\| \phi_2^T u(k) \widetilde{W}_{2,k} \right\| \right) \\ &\leq -2\eta_k \|r(k)\| \left\| \phi_1^T \widetilde{W}_{1,k} + \phi_2^T \widetilde{W}_{2,k} u(k) \right\| \\ &= -2\eta_k \|r(k)\| \|r(k+1) - k_v r(k) - \omega_1(k)\| \\ &= -2\eta_k \|r(k)\| \|r(k+1) - k_v r^2(k) - r(k) \omega_1(k)\| \end{aligned} \quad (5.27)$$

si  $\beta \|r(k+1)\| \geq \|r(k)\|$

$$\begin{aligned} &-2\eta_k \left\| r(k) \phi_1^T \widetilde{W}_{1,k} \right\| - 2\eta_k \left\| r(k) \phi_2^T \widetilde{W}_{2,k} u(k) \right\| \\ &\leq -\frac{2\eta_k}{\beta} \|r(k)\|^2 + 2\eta_k k_v \|r(k)\|^2 + \eta_k \|r(k)\|^2 + \eta_k \|\omega_1(k)\|^2 \end{aligned} \quad (5.28)$$

Usando  $0 < \eta \leq 1$ ,  $0 \leq \eta_k \leq \eta \leq 1$ ,  $\eta_k = \frac{\eta}{1 + \|\phi_1\|^2 + \|\phi_2 u\|^2}$

$$\begin{aligned} \Delta L_k &= \eta_k^2 r^2(k) (\|\phi_1\|^2 + \|\phi_2 u\|^2) - \frac{2\eta_k}{\beta} r(k)^2 + 2\eta_k k_v r(k)^2 + \eta_k r(k)^2 + \eta_k \omega_1^2(k) \\ &= -\eta_k \left[ \left( \frac{2}{\beta} - 2k_v - 1 \right) - \eta \frac{\|\phi_1\|^2 + \|\phi_2 u\|^2}{1 + \|\phi_1\|^2 + \|\phi_2 u\|^2} \right] r^2(k) + \eta_k \omega_1^2(k) \\ &\leq -\pi r^2(k) + \eta \omega_1^2(k) \end{aligned} \quad (5.29)$$

donde  $\pi = \frac{\eta}{1 + \kappa} \left[ \left( \frac{2\eta_k}{\beta} - 2k_v - 1 \right) - \frac{\kappa}{1 + \kappa} \right]$ ,  $\kappa = \max_k (\|\phi_1\|^2 + \|\phi_2 u\|^2)$ . Puesto que  $\frac{1}{1+k_v} \geq \beta \geq 1$ ,  $\left( \frac{2}{\beta} - 2k_v - 1 \right) > 1$ ,  $\pi > 0$

$$n \min (\tilde{w}_i^2) \leq V_k \leq n \max (\tilde{w}_i^2) \quad (5.30)$$

donde  $n \times \min (\tilde{w}_i^2)$  y  $n \times \max (\tilde{w}_i^2)$  son funciones  $\mathcal{K}_\infty$ , y  $\pi e^2(k)$  es una función  $\mathcal{K}_\infty$ ,  $\eta \zeta^2(k)$  es una función  $\mathcal{K}$ . De las ecuaciones (5.22) y (5.25) conocemos  $V_k$  es una función de  $e(k)$  y  $\zeta(k)$ , por eso  $V_k$  admite una función de Lyapunov ISS. la dinámica del error de identificación estable entrada estado. La "ENTRADA" corresponde al segundo termino de la ultima línea en la ecuación (5.29), *entonces* el modelado del error  $\zeta(k) = \varepsilon(k) + \mu(k)$ , el "ESTADO" corresponde al primer término de la ultima línea en la ecuación (5.29), *entonces* el error de identificación  $e(k)$ . Por que la "ENTRADA"  $\zeta(k)$  es acotada y la dinámica es ISS, el "ESTADO"  $e(k)$  es acotado.

Si  $\beta \|r(k+1)\| < \|r(k)\|$ ,  $\Delta L_k = 0$ .  $L_k$  es constante,  $W_{1,k}$ ,  $W_{2,k}$  son las constantes. Puesto que  $\|r(k+1)\| < \frac{1}{\beta} \|r(k)\|$ ,  $\frac{1}{\beta} < 1$ ,  $r(k)$  es acotado. ■

### 5.2.2. Aprendizaje con funciones pertenencia

Ahora consideramos una red neuronal multicapa(o perceptrones multicapa. MLP) que se representan como [26] para aproximar funciones no lineales  $f[x(k)]$  y  $g[x(k)]$  en la ecuación (5.3). La ecuación (5.17) se cambia como

$$\begin{aligned} \hat{f}[x(k)] &= W_{1,k} \phi_1 [V_{1,k} x(k)] \\ \hat{g}[x(k)] &= W_{2,k} \phi_2 [V_{2,k} x(k)] \end{aligned} \quad (5.31)$$

donde los pesos en la capa de salida son  $V_{1,k}, V_{2,k} \in R^{m \times n}$ , los pesos en la capa oculta son  $W_{1,k}, W_{2,k} \in R^{1 \times m}$ .  $m$  es la dimensión de la capa oculta,  $n$  es la dimensión del estado. El control de retroalimentación es entonces  $u(k)$  se puede definir de la manera siguiente

$$\begin{aligned} u(k) &= [W_{2,k} \phi_2 (V_{2,k} x(k))]^+ \\ [x_n^*(k+1) - W_{1,k} \phi_1 (V_{1,k} x(k)) + k_v r(x) - \lambda_1 e_n(k) - \dots - \lambda_{n-1} e_2(k)] \end{aligned} \quad (5.32)$$

El sistema de lazo cerrado llega a ser

$$r(k+1) = k_v r(k) + \tilde{f} + \tilde{g}u(k) \quad (5.33)$$

Simultáneamente a la ecuación (5.16)

$$\begin{aligned} \tilde{f} &= W_1^* \phi_1 [V_1^* x(k)] - W_{1,k} \phi_1 [V_{1,k} x(k)] - \tilde{f}_1 \\ \tilde{g}u(k) &= W_2^* \phi_2 [V_2^* x(k)] u(k) - W_{2,k} \phi_2 [V_{2,k} x(k)] u(k) - \tilde{g}_1 \end{aligned} \quad (5.34)$$

En el caso de dos variables independientes, una función suave  $f$  tiene una fórmula de Taylor de la forma

$$f(x_1, x_2) = \sum_{k=0}^{l-1} \frac{1}{k!} \left[ (x_1 - x_1^0) \frac{\partial}{\partial x_1} + (x_2 - x_2^0) \frac{\partial}{\partial x_2} \right]_0^k f + R_l \quad (5.35)$$

donde  $R_l$  recordando de la fórmula de Taylor. Si permitiéramos que  $x_1$  y  $x_2$  correspondan a  $W_1^*$ ,  $V_1^*$ ,  $x_1^0$  y  $x_2^0$  corresponda  $W_{1,k}$  y  $V_{1,k}$ ,

$$\begin{aligned} &W_1^* \phi_1 [V_1^* x(k)] - W_{1,k} \phi_1 [V_{1,k} x(k)] \\ &= \tilde{W}_{1,k} \phi_1 [V_{1,k} x(k)] + W_{1,k} \phi_1' [V_{1,k} x(k)] \tilde{V}_{1,k} x(k) + R_1 \end{aligned} \quad (5.36)$$

donde  $\phi_1'$  es la derivada de la función de activación no lineal  $\phi_1(\cdot)$  en un punto de  $V_{1,k} x(k)$ ,  $\tilde{W}_{1,k} = W_{1,k} - W_1^*$ ,  $\tilde{V}_{1,k} = V_{1,k} - V_1^*$ . Y

$$\begin{aligned} &W_2^* \phi_2 [V_2^* x(k)] u(k) - W_{2,k} \phi_2 [V_{2,k} x(k)] u(k) \\ &= \tilde{W}_{2,k} \phi_2 [V_{2,k} x(k)] u(k) + W_{2,k} \phi_2' [V_{2,k} x(k)] \tilde{V}_{2,k} x(k) u(k) + R_2 \end{aligned} \quad (5.37)$$

de donde  $\tilde{W}_{2,k} = W_{2,k} - W_2^*$ ,  $\tilde{V}_{2,k} = V_{2,k} - V_2^*$ . El sistema de lazo cerrado ecuación (5.33) es

$$r(k+1) = k_v r(k) + \omega_2(k) + \tilde{W}_{1,k} \phi_1 + W_{1,k} \phi_1' \tilde{V}_{1,k} x + \tilde{W}_{2,k} \phi_2 u + W_{2,k} \phi_2' \tilde{V}_{2,k} x u \quad (5.38)$$

donde  $\omega_2(k) = \tilde{f}_1 + \tilde{g}_1 + R_1 + R_2$ ,  $\|\omega_2(k)\|^2 \leq \bar{\omega}_2$ .

**Teorema 5.3** *Si usamos en neuro control adaptable ecuación (5.32) al control de la planta no lineal ecuación (5.3), la siguiente ley de actualización hace que el error de seguimiento  $r(k)$  sea acotado (estable en el sentido  $L_\infty$ )*

$$\begin{aligned} W_{1,k+1} &= W_{1,k} - \eta_k r(k) \phi_1^T, & V_{1,k+1} &= V_{1,k} - \eta_k r(k) r(k) \phi_1' W_{1,k}^T x^T(k) \\ W_{2,k+1} &= W_{2,k} - \eta_k r(k) u \phi_1^T, & V_{2,k+1} &= V_{2,k} - \eta_k r(k) u \phi_1' W_{2,k}^T x^T \end{aligned} \quad (5.39)$$

donde  $\eta_k$  satisfice

$$\eta_k = \begin{cases} \frac{\eta}{\|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x^T\| + \|\phi'_2 W_{2,k}^T x^T\|} & \text{si } \beta \|r(k+1)\| \geq \|r(k)\| \\ 0 & \text{si } \beta \|r(k+1)\| < \|r(k)\| \end{cases} \quad (5.40)$$

aquí  $1 \geq \eta > 0$ ,  $\frac{1}{1+k_v} \geq \beta \geq 1$ . El error promedio de la identificación satisfice

$$J = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T r^2(k) \leq \frac{\eta \bar{\omega}_2}{\pi} \quad (5.41)$$

donde  $\pi = \frac{\eta}{1+\kappa} \left[ 1 - \frac{\kappa}{1+\kappa} \right] > 0$ ,  $\kappa = \text{máx}_k \left( \|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x^T\| + \|\phi'_2 W_{2,k}^T x^T u\| \right)$ ,  
 $\bar{\omega}_2 = \text{máx}_k [\bar{\omega}_2^2(k)]$ .

**Demostación.** Seleccionamos una función de Lyapunov como

$$L_k = \|\tilde{W}_{1,k}\|^2 + \|\tilde{W}_{2,k}\|^2 + \|\tilde{V}_{1,k}\|^2 + \|\tilde{V}_{2,k}\|^2 \quad (5.42)$$

De la ley de actualización ecuación (5.39)

$$\begin{aligned} \Delta L_k &= \|\tilde{W}_{1,k} - \eta_k r(k) \phi_1\|^2 - \|\tilde{W}_{1,k}\|^2 + \|\tilde{W}_{2,k} - \eta_k r(k) \phi_2 u\|^2 - \|\tilde{W}_{2,k}\|^2 \\ &+ \|\tilde{V}_{1,k} - \eta_k r(k) \phi'_1 W_{1,k}^T x^T(k)\|^2 - \|\tilde{V}_{1,k}\|^2 + \|\tilde{V}_{2,k} - \eta_k r(k) \phi'_2 W_{2,k}^T x^T(k)^T u\|^2 - \|\tilde{V}_{2,k}\|^2 \\ &= \eta_k^2 r^2(k) \left( \|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x^T\| + \|\phi'_2 W_{2,k}^T x^T u\| \right) \\ &- 2\eta_k \|r(k)\| \left( \|\phi_1 \tilde{W}_{1,k}\| + \|\phi_2 \tilde{W}_{2,k} u\| + \|\tilde{V}_{1,k} \phi'_1 W_{1,k}^T x\| + \|\tilde{V}_{2,k} \phi'_2 W_{2,k}^T x u\| \right) \end{aligned} \quad (5.43)$$

Usando la ecuación (5.38), similar con el Teorema (5.3)

$$\begin{aligned} &-2\eta_k \|r(k)\| \left( \|\phi_1 \tilde{W}_{1,k}\| + \|\phi_2 \tilde{W}_{2,k} u\| + \|\tilde{V}_{1,k} \phi'_1 W_{1,k}^T x^T\| + \|\tilde{V}_{2,k} \phi'_2 W_{2,k}^T x^T u\| \right) \\ &\leq -2\eta_k \|r(k)\| r(k+1) - k_v r^2(k) - r(k) \omega_2(k) \end{aligned} \quad (5.44)$$

si  $\beta \|r(k+1)\| \geq \|r(k)\|$

$$\begin{aligned} &-2\eta_k \|r(k)\| \left( \|\phi_1 \tilde{W}_{1,k}\| + \|\phi_2 \tilde{W}_{2,k} u\| + \|\tilde{V}_{1,k} \phi'_1 W_{1,k}^T x^T\| + \|\tilde{V}_{2,k} \phi'_2 W_{2,k}^T x^T u\| \right) \\ &\leq -\frac{2\eta_k}{\beta} \|r(k)\|^2 + 2\eta_k k_v \|r(k)\|^2 + \eta_k \|r(k)\|^2 + \eta_k \|\omega_2(k)\|^2 \end{aligned} \quad (5.45)$$

Usando  $0 < \eta \leq 1$ ,  $0 \leq \eta_k \leq \eta \leq 1$ ,  $\eta_k = \frac{\eta}{\|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x\| + \|\phi'_2 W_{2,k}^T x u\|}$

$$\begin{aligned} \Delta L_k &= \eta_k^2 r^2(k) \left( \|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x^T\| + \|\phi'_2 W_{2,k}^T x^T u\| \right) \\ &\quad - \frac{2\eta_k}{\beta} r(k)^2 + 2\eta_k k_v r(k)^2 + \eta_k r(k)^2 + \eta_k \omega_2^2(k) \\ &= -\eta_k \left[ \left( \frac{2}{\beta} - 2k_v - 1 \right) - \eta \frac{\|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x\| + \|\phi'_2 W_{2,k}^T x^T u\|}{1 + \|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x\| + \|\phi'_2 W_{2,k}^T x^T u\|} \right] r^2(k) \\ &\quad + \eta_k \omega_2^2(k) \\ &\leq -\pi r^2(k) + \eta \omega_2^2(k) \end{aligned} \tag{5.46}$$

donde  $\pi = \frac{\eta}{1 + \kappa} \left[ \left( \frac{2\eta_k}{\beta} - 2k_v - 1 \right) - \frac{\kappa}{1 + \kappa} \right]$ ,  $\kappa = \max_k (\|\phi_1\|^2 + \|\phi_2 u\|^2 + \|\phi'_1 W_{1,k}^T x^T\| + \|\phi'_2 W_{2,k}^T x^T u\|)$ . Puesto que  $\frac{1}{1+k_v} \geq \beta \geq 1$ ,  $\left( \frac{2}{\beta} - 2k_v - 1 \right) > 1$ ,  $\pi > 0$

$$n \min(\tilde{w}_i^2) \leq V_k \leq n \max(\tilde{w}_i^2) \tag{5.47}$$

donde  $n \times \min(\tilde{w}_i^2)$  y  $n \times \max(\tilde{w}_i^2)$  son funciones  $\mathcal{K}_\infty$ , y  $\pi e^2(k)$  es una función  $\mathcal{K}_\infty$ ,  $\eta \zeta^2(k)$  es una función  $\mathcal{K}$ . De las ecuaciones (5.38) y (5.42) conocemos  $V_k$  es una función de  $e(k)$  y  $\zeta(k)$ , por eso  $V_k$  admite una función de Lyapunov ISS. La dinámica del error de identificación estable entrada estado. La "ENTRADA" corresponde al segundo término de la última línea en la ecuación (5.46), *entonces* el modelado del error  $\zeta(k) = \varepsilon(k) + \mu(k)$ , el "ESTADO" corresponde al primer término de la última línea en la ecuación (5.46), *entonces* el error de identificación  $e(k)$ . Por que la "ENTRADA"  $\zeta(k)$  es acotada y la dinámica es ISS, el "ESTADO"  $e(k)$  es acotado.

· Si  $\beta \|r(k+1)\| < \|r(k)\|$ ,  $\Delta L_k = 0$ .  $L_k$  es constante,  $W_{1,k}$ ,  $W_{2,k}$  son las constantes. Puesto que  $\|r(k+1)\| < \frac{1}{\beta} \|r(k)\|$ ,  $\frac{1}{\beta} < 1$ ,  $r(k)$  es acotado. La ecuación (5.29) se reescribe como

$$\Delta L_k \leq -\pi r^2(k) + \eta \omega_2^2(k) \leq \pi r^2(k) + \eta \bar{\omega}_2 \tag{5.48}$$

Sumando la ecuación (5.48) de 1 a  $T$ , y usando  $L_T > 0$  y  $L_1$  es una constante, obtenemos

$$\begin{aligned} L_T - L_1 &\leq -\pi \sum_{k=1}^T r^2(k) + T \eta \bar{\omega}_2 \\ \pi \sum_{k=1}^T r^2(k) &\leq L_1 - L_T + T \eta \bar{\omega}_2 \leq L_1 + T \eta \bar{\omega}_2 \end{aligned} \tag{5.49}$$

entonces la ecuación (5.41) se demuestra. ■

### 5.3. Simulación

Para esta simulación usaremos un sistema no lineal en tiempo discreto comparando con otros modelos.

Como [43]:

$$y(k+2) = \frac{[y(k) * y(k) + y(k+1) * y(k+1) + 2.5 * \tanh(y(k)) + 1] z(k)}{(y(k) * y(k) + y(k+1) * y(k+1) + 1)} + u(k) \quad (5.50)$$

$$z(k+1) = b1 * y(k+1) + b2 * z(k)$$

Como [16]:

$$y(k+2) = \frac{1.5 * [y(k) * y(k+1)]}{(y(k) * y(k) + y(k+1) * y(k+1) + 1) + 0.35 * \sin(y(k) + y(k+1))} + u(k) \quad (5.51)$$

Como [27]:

$$y(k+2) = \frac{(y(k)^2 + y(k+1)^2 + 2.5 * \tanh(y(k) + 1) * u(k))}{(y(k) * y(k) + y(k+1) * y(k+1) + 1)} + z(k) \quad (5.52)$$

donde  $y(k)$  es la posición a manipular,  $u(k)$  es el control del torque,  $T$  es el tiempo de muestra. Si definimos  $x_1(k) = y(k)$ ,  $x_2(k) = y(k+1)$ , las ecuaciones (5.50), (5.51) y (4.7) se pueden transformar en la ecuación (5.3)

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= f[x(k)] + g[x(k)] u(k) \end{aligned} \quad (5.53)$$

donde  $x(k) = [x_1(k), x_2(k)]^T$ ,  $f[x(k)] = 2(1-T)x_2(k) + (2T-1)x_1(k) + 10T^2 \sin x_1(k)$ ,  $g[x(k)] = 1$ . El error de seguimiento se escribe como

$$r(k) = e_2(k) + \lambda_1 e_1(k) \quad (5.54)$$

donde  $e_2(k) = x_2(k) - x_2^*(k) = y(k+1) - y^*(k+1)$ ,  $e_1(k) = x_1(k) - x_1^*(k) = y(k) - y^*(k)$ ,  $y^*(k)$  es la trayectoria deseable. Si nosotros seleccionamos  $\lambda_1 = 0.5$ ,  $T = 0.01$ ,  $y^*(k) = \frac{\pi}{2} \sin\left(\frac{\pi}{5}k\right)$ . Por que  $g[x(k)] = 1$ , la red neuronal difusa se cambia como

$$\hat{f}[x(k)] = W_{1,k} \phi_1[V_{1,k} x(k)] \quad (5.55)$$

de donde  $V_{1,k} \in R^{20 \times 2}$ ,  $W_{1,k} \in R^{1 \times 20}$ . Entonces el control adaptable  $u(k)$  puede ser

$$u(k) = x_2^*(k+1) - W_{1,k} \phi_1 [V_{1,k} x(k)] + k_v r(k) - \lambda_1 e_2(k) \quad (5.56)$$

donde  $k_v = -0.5$ . La actualización de los pesos como

$$\begin{aligned} W_{1,k+1} &= W_{1,k} - \eta_k r(k) \phi_1^T [V_{1,k} x(k)] \\ V_{1,k+1} &= V_{1,k} - \eta_k r(k) \phi_1' W_{1,k}^T x^T(k) \end{aligned} \quad (5.57)$$

donde  $\eta_k$  satisface que

$$\eta_k = \begin{cases} \frac{\eta}{1 + \|\phi_1\|^2 + \|\phi_1' W_{1,k} x\|} & \text{si } \beta \|r(k+1)\| \geq \|r(k)\| \\ 0 & \text{si } \beta \|r(k+1)\| < \|r(k)\| \end{cases} \quad (5.58)$$

donde  $\eta = 1$ ,  $\beta = 2$ . El sistema de lazo cerrado se deriva de las ecuaciones (5.53), (5.56)

$$y(k+2) = f[x(k)] - \hat{f}[x(k)] + y^*(k+2) + k_v r(k) - \lambda_1 e_2(k) \quad (5.59)$$

Puesto que  $e_2(k) = e_1(k+1)$  y  $r(k) = e_2(k) + \lambda_1 e_1(k)$ , de manera tenemos

$$r(k+1) = r(k) + \omega(k) \quad (5.60)$$

donde  $r(k) = y(k+1) - y^*(k+1) + \lambda_1 y[k] - y^*(k)$ ,  $\omega(k) = f[x(k)] - \hat{f}[x(k)]$ .

Las simulaciones siguientes nos describen el sistema comparando con varios modelos descritos anteriormente ecuaciones (5.51), (5.50), (5.52).

Este modelo corresponde a Chen Figura (5.1). Este modelo corresponde a Narendra Figura (5.2). Este modelo corresponde a Wen Figura (5.3).

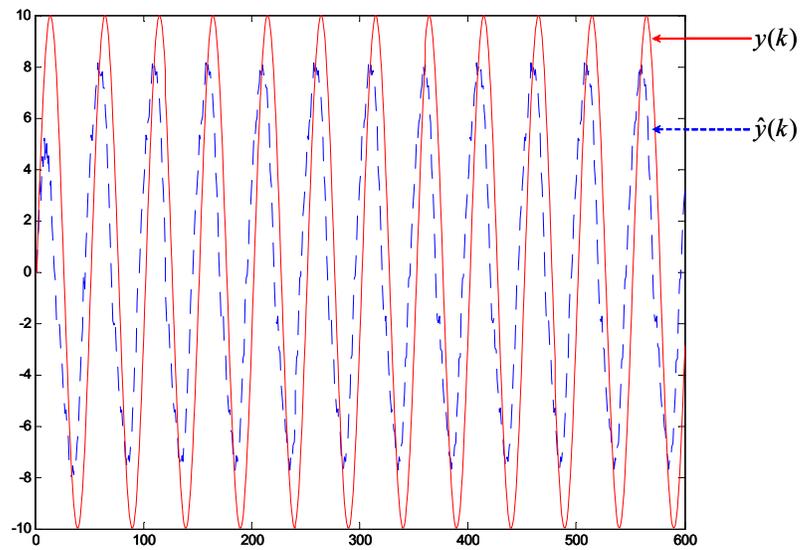


Figura 5.1: Identificación del sistema no lineal para el modelo Chen

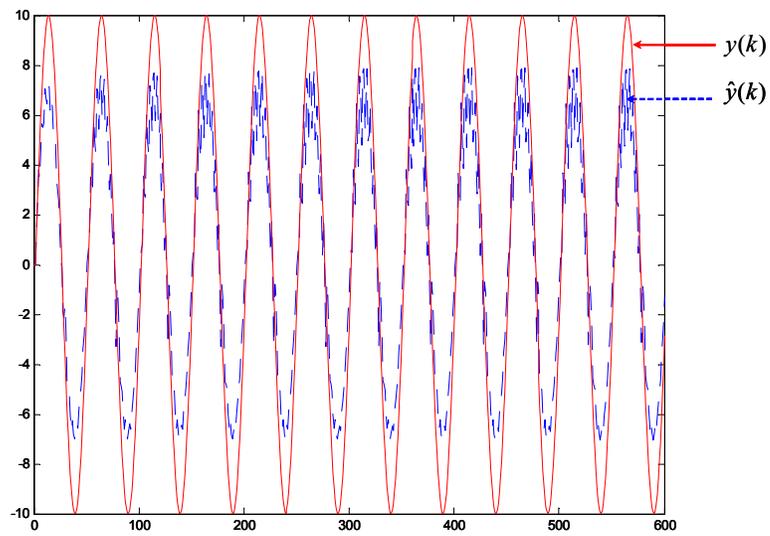


Figura 5.2: Identificación del sistema no lineal para el modelo Narendra

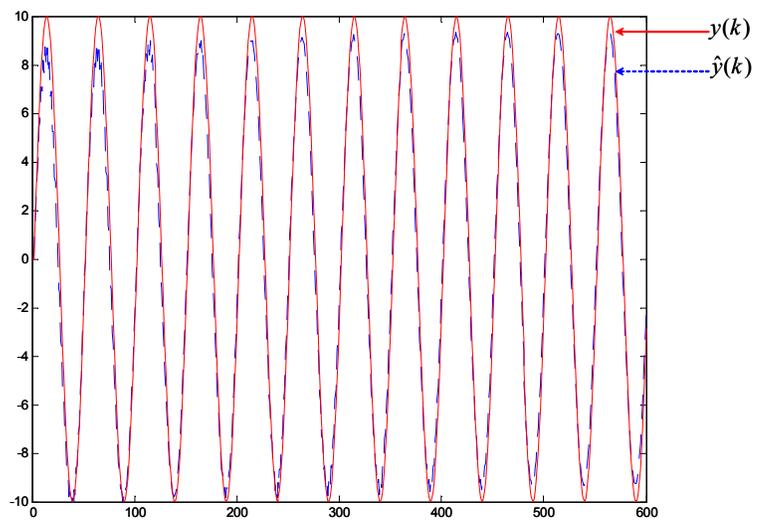


Figura 5.3: Identificación del sistema no lineal para el modelo Wen



# Capítulo 6

## Conclusiones

- Se propuso el modelo dinámico con un algoritmo de aprendizaje donde se aplican en varios problemas temporales incluyendo un modelo NARMA. La función de comparación son comportamientos con una sucesión estática y sistemas dinámicos y algunos modelos recurrentes difusos existentes esto se puede observar en las simulaciones donde se pueden considerar sistemas de identificación eficientes.
- En esta tesis estudiamos el modelado de un sistema no lineal con redes neuronales difusas usando un algoritmo de aprendizaje estable. Aproximando a una entrada estado estable, concluimos que las técnicas comúnmente usadas de robusticidad, como un método de modificación sigma, de donde no necesariamente para la ley del gradiente descendiente. Posteriormente se trabajara haciendo aprendizajes con funciones pertenencia conocidas y luego para funciones pertenencia desconocidas.
- La identificación neuro difuso y control en sistemas estáticos y dinámicos, también se hicieron comparaciones generales en sistemas difusos y redes neuronales, también usamos la estabilidad estado entrada (ISS) para aproximar la función a estabilizar. Además se uso la estructura tipo mamdani y takagi sugeno kang (TSK) para la estática difusa en redes neuronales
- Finalmente tenemos que las redes neuronales difusas dinámicas para sistemas no lin-

eales de control adaptable la función a identificar no se conoce y se parte haciendo uso de neuro difuso así notamos que el sistema y la identificación comparando con otros modelos en algunos casos decimos que fue fructífera y en algunos casos segun los valores declarados hacían el sistema inestable para otro modelo, concluimos que el método de identificación neuro difuso en sistemas no lineales de control adaptable es buena.

# Bibliografía

- [1] A.Albert, *Regression and the Moore-Penrose Pseudoinverse*, Academic Press, 1972.
- [2] F.C.Chen and H.K.Khalil, Adaptive control of nonlinear systems using neural networks, *Int.J.Control*, Vol.55,No.6, 1299-1317,1992
- [3] F.C.Chen and H.K.Khalil, Adaptive control of a class of nonlinear discrete-time systems using neural networks, *IEEE Trans. Automatico Control*, Vol.40, No.5, 791-801, 1995.
- [4] G.Cybenko, Approximation by Superposition of Sigmoidal Activation Function, *Math.Control, Sig Syst*, Vol.2, 303-314, 1989
- [5] C.F.Juang, A TSK-Type recurrent fuzzy network for dynamic systems procesing by neural network and genetic algorithm, *IEEE Trans. Fuzzy Syst.*, Vol. 10, 155-170, 2002.
- [6] P.A. Mastorocostas and J.B. Theocharis, A recurrent fuzzy-neural model for dinamic systems identification, *IEEE Trans. Syst., Man, Cybern. B*, Vol. 32, 179-190,2002.
- [7] Nguyen, D. H. and Widrow, B., "Neural Networks for Self-Learning Control Systems," *IEEE Control Systems Magazine*, vol. 10 pp. 18-23, 1990.
- [8] Tsoi, A. C. and Back, A. D., "Locally Recurrent Globally Feedforward Networks: A Critical Review of Architectures", *IEEE Trans. on Neural Networks*, Vol.5, No. 2, March 1994, pp. 229-239.
- [9] Haykin, S., "*Neural Networks, A Comprehensive Foundation*", Prentice Hall U.S., 2nd Ed. 1999.

- [10] Hilera, J. R. y Martínez V. J., "Redes Neuronales Artificiales. Fundamentos Modelos y Aplicaciones", Ed. RA-MA, 1995.
- [11] Martin B.B. y Sanz M. A. "Redes Neuronales y Sistemas Difusos", Ra-Ma, 2nd Ed 2001.
- [12] Li-Xin W. "A course in Fuzzy Systems and Control", Ed. prentice Hall, 1997.
- [13] M. Brown. y C.J. Harris, "Neurofuzzy Adaptive Modelling and Control", Prentice Hall, New York, 1994.
- [14] C.T.Lin and G.Lee, Neural fuzzy systems: *A neural-fuzzy synergism to intelligent systems*, Prentice-Hall Inc., NJ, 1996.
- [15] W.C.Chan, C.W.Chan, K.C.Cheung and Y.Wang, Modeling of nonlinear stochastic dynamical systems using neurofuzzy networks, *38th IEEE Conference on Decision and Control*, 2643-2648, Phoenix, USA, 1999.
- [16] D.S.Chen and R.C.Jain, A robust back propagation learning algorithm for fuunction approximation, *IEEE Trans.Neural Networks*, Vol5, No3, 1994.
- [17] M.Y.Chen and D.A.Linkensm, A systematic neuro-fuzzy modeling framework with application to material property prediction, *IEEE Trans. Syst., Man, Cybern. B*, Vol.31, 781-790, 2001.
- [18] B.Egardt, *Stability of Adaptive Controllers*, Lecture Notes in Control and Information Sciences, Vol.20, Springer-Verlag, Berlin, 1979.
- [19] S.I.Horikawa, T.Furuhashi and Y.Uchikawa, On fuzzy modeling using fuzzy neual networks with the back-propagation algorithm, *IEEE Trans. Neural NetiDorks*, Vol. 3 No.5, 801-806, 1992.
- [20] k.Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks*, Vol.4 No.5,251-257,1991.

- [21] P.A.Ioannou and J.Sun, *Robust Adaptive Control*, Prentice-Hall, Inc, Upper Saddle River: NJ, 1996.
- [22] S.Jagannathan and F.L.Lewis, Identification of Nonlinear Dynamical Systems Using Multilayered Neural Networks, *Automatica*, Vol.32, No.12, 1707-1712, 1996.
- [23] Z.P.Jiang and Y.Wang, Input-to-State Stability for Discrete-Time Nonlinear Systems, *Automatica*, Vol.37, No.2, 857-869, 2001.
- [24] Y.G.Leu, T.T.Lee and W.Y.Wang, Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems, *IEEE Trans. Syst., Man, Cybern. B*, Vol. 29, 583-591, 1999.
- [25] C.T.Lin, A neural fuzzy control system with structure and parameter learning, *Fuzzy Sets and Systems*. Vol.70, 183-212, 1995.
- [26] F.L.Lewis, A.Yesildirek and K.Liu, Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance, *IEEE Trans. Neural Networks*, Vol.7, No.2, 388-399, 1996.
- [27] K.S.Narendra and S.Mukhopadhyay, Adaptive Control Using Neural Networks and Approximate Models, *IEEE Trans. Neural Networks* , Vol.8, No.3, 475-485, 1997.
- [28] H.H.Tsai and P.T.Yu, On the optimal design of fuzzy neural networks with robust learning for function approximation, *IEEE Trans. Syst., Man, Cybern. B*, Vol.30, 217-223, 2000.
- [29] S.Wu and M.J.Er, Dynamic fuzzy neural networks- a novel approach to function approximation, *IEEE Trans.Syst, Man, \_ Cybern. B*, Vol.30, 358-364, 2000.
- [30] C.H.Wang, H.L.Liu and C.T.Lin, Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm, *IEEE Trans. Syst., Man, Cybern, B*, Vol.31, 467-475, 2001.

- [31] Y.G.Leu, T.T.Lee and W.Y.Wang, Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems, *IEEE Trans. Syst., Man, Cybern. B*, Vol. 29, 583-591, 1999.
- [32] C.H.Wang, H.L.Liu and C.T.Lin, Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm, *IEEE Trans. Syst., Man, Cybern. B*, Vol.31, 467-475, 2001.
- [33] L.X.Wang, *Adaptive Fuzzy Systems and Control*, Englewood Cliffs NJ: Prentice-Hall, 1994.
- [34] W.Y.Wang, T.T.Lee, C.L.Liu and C.H.Wang, Function approximation using fuzzy neural networks with robust learning algorithm, *IEEE Trans. Syst., Man, Cybern. B*, Vol.27, 740-747, 1997.
- [35] W.Y.Wang, Y.G.Leu and C.C.Hsu, Robust adaptive fuzzy neural control of nonlinear dynamical systems using generalized projection updated law and variable structure controller, *IEEE Trans. Syst., Man, Cybern. B*, Vol.31, 140-147, 2001.
- [36] G.A. Rovithakis and M.A. Christodoulou, Adaptive Control of Unknown Plants Using Dynamical Neural Networks, *IEEE Trans. on Syst., Man and Cybern.*, Vol. 24, 400-412, 1994.
- [37] E.B. Kosmatopulos, M.M. Polycarpou, M.A. Christodoulou and P.A. Ioannpu, High order neural networks structures for identification of dynamical systems, *IEEE Trans. Neural Network*, Vol. 6, No. 2, 422-421, 1995.
- [38] A.S. Poznyak, W. Yu, E. N. Sanchez and J.P. Perez, Nonlinear Adaptive Trajectory Tracking Using Dynamical neural Networks, *IEEE Trans. on Neural Networks*, Vol. 10, No. 5, 1402-1411, 1999.
- [39] Wen Yu and Alexander S. Poznyak, Indirect Adaptive control via parallel dynamic neural networks, *IEE Proceeding - Control Theory and Applications*, Vol. 146, No. 1, 25-30, 1999.

- [40] Wen Yu, Xiaou Li, Some stability properties of dynamic neural networks, *IEEE Trans. Circuits and Systems*, part I, Vol. 48, No. 2, 256-159, 2001.
- [41] W.Yu and X. Li, Some stability properties of dynamic neural networks, *IEEE Trans. Circuits and Systems*, Part I, Vol.48, No.1, 256-259, 2001.
- [42] W.Yu and X. Li, Some new results on system identification with dynamic neural networks, *IEEE Trans. Neural Networks*, Vol.12, No.2, 412-417,
- [43] Wen Yu, Fuzzy Identification Using Fuzzy Neural Networks with Stable Learning Algorithms, *IEEE Trans. on Fuzzy Systems*, accepted for publication
- [44] Wen Yu, Rigoberto Toxqui, Stable Fuzzy Identification Using Recurrent Fuzzy Neural Networks"en el *IASTED International Conference Neural Networks and Computational Intelligence*, 2003, Cancun, México