CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Control Automático

Título de la Tesis:

Análisis y diseño de redes neuronales CMAC para la identificación y control de sistemas no lineales

Tesis que presenta:

M. en C. Floriberto Ortiz Rodríguez[1]

Para obtener el grado de:

Doctor en Ciencias
En la Especialidad de Control Automático

Directores de Tesis:

Dr. Wen Yu Liu
Dr. Marco A. Moreno Armendáriz

México DF., Octubre del 2008

[1] **Becario 176273 del CONACyT.**

Índice general

Índice de figuras

0.1. Notación

$$
\begin{aligned}
s, y &\triangleq \textit{entrada} - \textit{salida red} \\
x &\triangleq \textit{estados} \\
\mathbb{R}^n &\triangleq \textit{espacio dimensional} \\
h, i, j, o, p, q, r &\triangleq \textit{subíndices} \\
f, g, \phi, \varphi &\triangleq \textit{funciones} \\
w, \theta &\triangleq \textit{pesos} \\
a &\triangleq \textit{memoria activada} \\
\mu &\triangleq \textit{función Gaussiana} \\
mf &\triangleq \textit{función de pertenencia} \\
m &\triangleq \textit{pertenencia recurrente} \\
t, k &\triangleq \textit{tiempo } (\textit{continuo}, \textit{discreto}) \\
m &\triangleq \textit{No pertenencia} \\
q &\triangleq \textit{cuantización} \\
n_a &\triangleq \textit{elementos asociación} \\
h &\triangleq \textit{hipercubos activados} \\
n_r &\triangleq \textit{elementos resolución} \\
A &\triangleq \textit{espacio asociación} \\
T &\triangleq \textit{espacio receptivo} \\
r_s &\triangleq \textit{resolución} \\
\lambda_{\text{mín}}(A) &\triangleq \textit{valor propio mas pequeño de la matriz } A \\
\lambda_{\text{máx}}(A) &\triangleq \textit{valor propio mas grande de la matriz } A \\
x_i &\triangleq \textit{el } i - \textit{ésimo elemento del vector } x \\
\|x\| &\triangleq \textit{norma del vector } x \\
a_{ij} &\triangleq \textit{el } ij - \textit{ésimo elemento de la matriz } A
\end{aligned}
$$

Capítulo 1

Introducción

Los sistemas actuales a controlar se han vuelto muy complejos, la mayoría de ellos presentan algún grado de no linealidad, pueden ser variantes e invariantes en el tiempo, presentan incertidumbres en sus entradas y en su estructura. Las incertidumbres en su entrada son causadas por la imprecisión al medir el valor de sus parámetros ó desconocimiento de los mismos, mientras que las incertidumbres en su estructura se refieren a las dinámicas no modeladas, como fricción no lineal, acoplamiento de engranes, ruido en los sensores, perturbaciones externas, etc., este tipo de fenómenos no pueden ser tratados por las técnicas de control clásico [18]. Las consideraciones sobre sensibilidad son importantes en el diseño de sistemas de control, ya que todos los elementos físicos tienen la propiedad de cambiar con el ambiente y con el tiempo, no se pueden considerar los parámetros de un sistema de control completamente estacionarios durante toda su vida de operación, todos los sistemas físicos están sujetos a señales externas ó ruido durante su funcionamiento. Una perturbación es una señal que tiende a afectar negativamente el comportamiento de la salida de un sistema. Si la perturbación se genera dentro del sistema se denomina interna, en tanto que una perturbación externa se produce fuera del sistema, es decir, es una señal de entrada. En general, un buen sistema de control debe ser insensible a la variación de los parámetros pero capaz de responder a los comandos ó datos de entrada. El efecto de la retroalimentación sobre el ruido y la perturbación depende en mayor medida en que parte del sistema se presentan las

señales externas, en la mayoria de las ocasiones la realimentación puede reducir los efectos del ruido y las perturbaciones en el desempeño del sistema.

Uno de los métodos que más interes a despertado en la teoría de control de sistemas no lineales que pueden resolver los problemas mencionados anteriormente es el control inteligente, el cuál incorpora técnicas como las redes neuronales artificiales y los sistemas difusos. Las redes neuronales artificiales son un tipo especial de estructura matemática las cuáles se basan en el modelo biológico de las neuronas del cerebro. Una red neuronal tiene la propiedad de ajustar automáticamente sus parámetros mediante una regla llamada algoritmo de aprendizaje, generalmente este algoritmo se basa en la retropropagación del error, así la red puede aproximar una función no lineal con una gran exactitud. Esta notable característica permite que la red sea utilizada para identificar y controlar procesos complejos que son completamente ó parcialmente desconocidos, además puede compensar incertidumbres en las entradas del sistema. Una nueva topología de red neuronal propuesta por James Albus en 1976 llamada ***C****erebellar* ***M****odel* ***A****rticulatión* ***C****ontroller,* cuya traducción es: modelo del cerebelo para el control de la articulación ó CMAC por su acrónimo en inglés [1], [2]. El modelo biológico de la neurona se encuentra en el cerebelo, el cuál se encarga de controlar y coordinar todos los movimientos de las articulaciones en el ser humano, es capaz de controlar millares de músculos para producir un movimiento coordinado. La red CMAC es la representación artificial de este modelo y ha sido utilizado ampliamente para control de sistemas en lazo cerrado de sistemas dinámicos complejos principalmente en aplicaciones en tiempo real debido a su rápido proceso de aprendizaje, la red CMAC es una alternativa a las clásicas redes como el perceptrón multicapa (Multi Layer Perceptron, MLP).

Otra herramienta del control inteligente son los sistemas difusos, los cuáles tienen la particularidad de agregar conocimiento apriori y la experiencia de un experto acerca del sistema y de la estrategia de control. Una limitación importante que presentan los controladores difusos convencionales es que el número de reglas se incrementa exponencialmente al incrementarse el número de variables involucradas, por lo que se generan reglas redundantes y el diseño se complica, a esta limitante se le conoce como *.el problema de la dimensionalidad "*, de tal forma que con n variables de entradas y m funciones de pertenencia para cada variable,

se necesitan m^n reglas para construir la base de reglas del sistema difuso completo. Cuando las variables de entrada n se incrementan, la base de reglas aumenta rápidamente, esto hace que el controlador difuso sea difícil de implementar ya que el espacio de memoria requerido crece exponencialmente. Uno de los métodos desarrollados para resolver este problema fue propuesto por Raju, Zhou and Kisner [24], que consiste en diseñar una estructura jerárquica con un número de sistemas de baja dimensionalidad conectados en niveles jerárquicos como se observa en la figura 1.1. La estructura jerárquica tiene la propiedad de que el número total de reglas para los sistemas difusos se incrementa linealmente con respecto al número de variables de entrada. Por ejemplo si se definen m conjuntos difusos para cada variable de entrada, cada subsistema difuso requiere solo de m^2 reglas y el número total de reglas es $(n-1)m^2$, el cuál tiene un crecimiento lineal con respecto al número de variables de entrada n, esto representa una gran ventaja con respecto a los sistemas difusos convencionales.

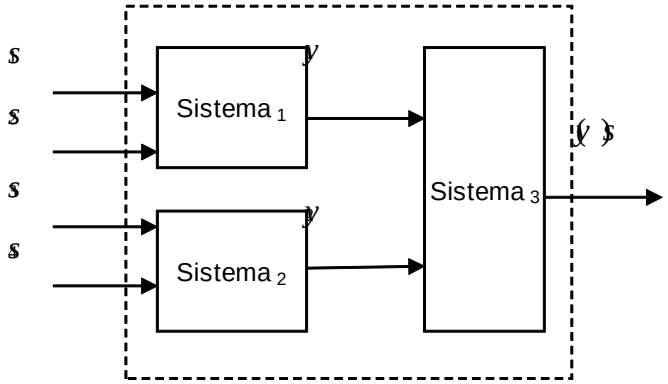

Figura 1.1: Estructura jerárquica.

Un sistema de control automático presenta un control conectado a una planta y una retroalimentación de las salidas ó de sus estados para compararlo con la señal de referencia, lo que puede llevar a modificar la dinámica del sistema original en lazo abierto y hacerlo inestable. En 1892 A. M. Lyapunov [3] presentó dos métodos para determinar la estabilidad de los sistemas dinámicos descritos mediante ecuaciones diferenciales ordinarias. El primer método se compone de todos los procedimientos en los cuales se usa la forma explícita de

la solución de las ecuaciones diferenciales para el análisis. En cambio, el segundo método no requiere de las soluciones de las ecuaciones diferenciales, es decir, mediante el segundo método de Lyapunov se determina la estabilidad de un sistema no lineal sin resolver las ecuaciones de estado, lo cuál resulta en un método más sencillo. Uno de los criterios, es la estabilidad de los puntos de equilibrio del sistema en el sentido de Lyapunov. Un punto de equilibrio es estable si todas las soluciones que inician cerca de este punto permanecen cerca, de otra forma es inestable. Es asintóticamente estable si todas las soluciones que empiezan cerca del punto de equilibrio, no solo permanecen cerca, si no que ademas tienden al punto de equilibrio en un tiempo muy grande. De manera general el método de Lyapunov es para sistemas autónomos (independiente del tiempo) y no autónomos (dependiente del tiempo) sin perturbaciones en la entrada del tipo $x\left(k+1\right)=f\left[k,x\left(k\right),0\right]$.

Otro criterio de estabilidad muy particular empleado en esta tesis es la estabilidad $entrada-estado$ (ISS por sus siglas en ingles), la cuál es una extensión natural de la estabilidad de Lyapunov para sistemas no lineales con entradas y cuyas salidas son los estados. Este enfoque estudia la dependencia de la trayectoria de los estados de los sistemas no lineales en tiempo continúo y discreto sobre la magnitud de las entradas $u\left(k\right)$, las cuáles pueden representar variables de control ó perturbaciones para sistemas con estructura $x\left(k+1\right)=f\left[k,x\left(k\right),u\left(k\right)\right]$, es decir, un sistema es $ISS-estable$ si la trayectoria de cada uno de los estados que se genera a partir de un control acotado permenece acotado y la trayectoria eventualmente llega a ser pequeña si la señal de entrada es pequeña independientemente del estado inicial.

En esta tesis se realiza el diseño de un identificador y un controlador neurodifuso para sistemas no lineales que combinan las ventajas de las redes neuronales CMAC y los sistemas difusos con estructura jerárquica y jerárquica-recurrente, se realiza el análisis de estabilidad del algoritmo de aprendizaje de retropropagación del error para las redes CMAC difusas mediante el criterio de estabilidad entrada-estado conocido como $ISS-Lyapunov$. Los siguientes acrónimos son útilizados en toda la tesis: red neuronal CMAC difusa (FCMAC), red neuronal recurrente CMAC difusa (RFCMAC), red neuronal jerárquica CMAC difusa (HFCMAC) y por último red neuronal jerárquica recurrente CMAC difusa (HRFCMAC).

1.1. Motivación

Las redes neuronales artificiales presentan la propiedad de ajustar sus pesos para tener un comportamiento similar a un sistema no lineal desconocido a ser aproximado, esta característica es utilizada para identificar y controlar procesos complejos y compensar las incertidumbres presentes en las entradas del sistema, esta herramienta matemática puede ser combinada con otra técnica de control inteligente que son los sistemas difusos que agregan conocimiento humano y experiencia apriori acerca del comportamiento del sistema y de la estrategia de control, dadas por un experto en forma de reglas de inferencia con estructura *si antecedente entonces consecuente,* el objetivo de combinar ambas técnicas es aprovechar las ventajas que presenta cada uno de ellos. Algunos de los casos en donde es necesario utilizar sistemas neuro-difusos ya sea para la identificación o control de sistemas no lineales son los siguientes:

- En procesos no lineales multivariables, que presentan un gran número de variables de entrada y de salida, en los cuáles solo se pueden conocer los datos de entrada y de salida del sistema.
- Cuando es necesario introducir el conocimiento a priori de un "experto" en el esquema de control, obtenido a partir de su experiencia y conocimiento práctico de la planta.
- Cuando se desea identificar y controlar plantas cuyos parámetros presentan algún grado de imprecisión e incertidumbre, en donde se desconoce parcial o completemente el modelo matemático del sistema por ser demasiado complejo.

Sin embargo las limitaciones causadas por el problema de la dimensionalidad en los sistemas neuro-difusos convencionales, sugieren una nueva metodología para el diseño de identificadores y controladores inteligentes a través del uso de una red FCMAC con estructura jerárquica. Los sistemas neuro-difusos jerárquicos evitan el crecimiento del número de reglas de inferencia de forma exponencial, solo lo hace de manera lineal, lo que evita un incremento en el espacio de memoria requerido para su implementación, esta característica hace que

la red FCMAC sea fácil de realizar principalmente en aplicaciones de tiempo real, además que el ajuste de los peso sólo se realiza en algunas neuronas activadas por las entradas, esto ocasiona que el algoritmo de aprendizaje para la actualización de los pesos de la red sea mucho mas rápido.

1.2. Objetivo de la tesis

El objetivo general de la tesis se enfoca en el diseño de una topologia de red neuronal artificial llamada FCMAC para la identificación y control de sistemas no lineales, se proponen dos tipos de redes FCMAC sin retroalimentación para atacar problemas estáticos y con retroalimentación para sistemás dinámicos. Además de que presentan una estructura jerárquica para minimizar los requerimientos de memoria, todos estos esquemas presentan un algoritmo de aprendizaje de la red entrada-estado estable.

1.2.1. Objetivos particulares

Los objetivos particulares se dividen principalmente en dos aspectos:

- El primero es realizar una identificación para sistemas estáticos no lineales mediante la integración de sistemas difusos y redes neuronales artificiales generalmente llamados sistemas neuro-difusos, basados en una topología de red conocida como red neuronal FCMAC, a partir de esta topología se hace una extención a redes recurrentes FCMAC con retroalimentación local en una de las capas de las red, con retroalimentación global entre la capa de entrada y de salida y con una combinación de ambas retroalimentaciones tanto local como global, todas estas retroalimentaciones para resolver los problemas de identificación en los sistemas dinámicos. Tanto a las redes estáticas como dinámicas FCMAC se presentan con una estructura jerárquica para resolver el problema de la dimensionalidad.

- El segundo objetivo es realizar un control adaptable indirecto mediante la misma topologia de red utilizada en la identificación: la red recurrente FCMAC con estruc-

tura jerárquica de baja dimensionalidad para resolver el problema del incremento de reglas presentes en controladores neuro-difusos convencionales. En la identificación y control mediante las redes FCMAC se realiza el análisis de estabilidad del algoritmo de aprendizaje tipo retro-propagación del erros para el ajuste de los pesos de la red, el cuál se basa en el segundo método de Lyapunov.

1.3. Estructura de la tesis

El trabajo de tesis titulado *"**Análisis y diseño de redes neuronales CMAC para la identificación y control de sistemas no lineales** "* está dividido en cinco capitulos:

El **capítulo uno** menciona los antecedentes de las redes neuronales artificiales, los diferentes conceptos de estabilidad y la motivación que se tuvo para realizar la identificación y control de sistemas no lineales mediante sistemas neuro-difusos, en particular la red neuronal CMAC recurrente con estructura jerárquica. Se menciona además los objetivos generales y particulares planteados en este trabajo de tesis.

El **capítulo dos** hace una breve reseña sobre el funcionamiento de los sistemas neuronales biológicos, la estructura y el proceso de aprendizaje que hace posible su interelación con el medio que lo rodea, se aborda de manera particular la anatomía del cerebelo, el cuál se encarga de coordinar y ejecutar de manera sincronizada todos los movimientos de las articulaciones del cuerpo humano. Del estudio de la anatomía del cerebelo parte un modelo matemático conocido como red neuronal CMAC difusa, del cuál se hace una breve descripción de su arquitecura y principio de funcionamiento.

El **capítulo tres** estudia las características más importantes de las redes neuronales y de los sistemas difusos como aproximadores de funciones y para la identificación de sistemas no lineales. En este capítulo se detalla el funcionamiento de la red neuronal CMAC y sus distintas modificaciones como son la red CMAC difusa, recurrente y con estructura jerárquica. Además de su aplicación en la identificación de sistemas no lineales en tiempo discreto. En la Identificación de los sistemas discretos se realiza el análisis de estabilidad entrada-estado del algoritmo de aprendizaje de la red FCMAC, el cuál se basa en la retropropagación del

error.

El **capítulo cuatro** estudia la capacidad de las redes neuronales CMAC para controlar sistemas, en este capítulo se trabaja con sistemas en tiempo continuo. Se desarrolla un control adaptable indirecto mediante la estructura de la red jerárquica FCMAC, se realiza el análisis de estabilidad mediante el segundo método de *Lyapunov* para demostrar la convergencia de los pesos en el algoritmo de aprendizaje de la red.

El **capítulo cinco** presenta las conclusiones del trabajo y se propone el estudio de otros tópicos de interés relacionados con los sistemas neuro-difusos CMAC. En este capitulo se mencionan las diferentes aplicaciones de la red CMAC, principalmente en sistemas electromecánicos subactuados y aplicaciones en tiempo real.

1.4. Publicaciones

1.4.1. Revista

1. Wen Yu, Marco A. Moreno-Armendariz, Floriberto Ortiz Rodriguez, System identification using hierarchical fuzzy neural networks with stable learning algorithm, **Journal of Intelligent & Fuzzy Systems**, Vol.18, No.2, 171-183, 2007.

2. Wen Yu, Floriberto Ortiz Rodríguez, Marco A. Moreno-Armendariz, Nonlinear systems identification via two types of recurrent fuzzy CMAC, **Neural Processing Letters**. Vol.28, No.1, 49-62, 2008.

3. Wen Yu, Floriberto Ortiz Rodríguez, Marco A. Moreno-Armendariz, Hierarchical fuzzy CMAC for nonlinear systems modeling, **IEEE Transactions on Fuzzy Systems**. Vol.16, No.5, 1302-1314, 2008.

4. Floriberto Ortiz Rodriguez, Wen Yu, Marco A. Moreno-Armendariz, Stable adaptive control with hierarchical fuzzy CMAC, **Journal Information Sciences**, Special Issue: Prediction, Control and Diagnosis using Advanced Neural Computations, aceptado para su publicación.

1.4.2. Congresos internacionales

1. Floriberto Ortiz Rodriguez, Wen Yu, Marco A. Moreno-Armendariz, System identification using hierarchical fuzzy CMAC neural networks, Computational Intelligence-ICIC2006, Srpinger-Verlag, **Lecture Notes in Computer Science**, LNAI 4114, 230-235, 2006.

2. Floriberto Ortiz Rodriguez, Wen Yu, Marco A.Moreno-Armendariz, Recurrent fuzzy CMAC in hierarchical form for dynamic system identification, **2007 American Control Conferences**, ACC'07, New York, USA, 5706-5711, 2007.

3. Floriberto Ortiz, Wen Yu, Marco Moreno-Armendariz, Xiaoou Li, Recurrent Fuzzy CMAC for Nonlinear System Modeling, Advances in Neural Networks -ISNN 2007, Srpinger-Verlgag, **Lecture Notes in Computer Science**, LNCS 4491, 487-495, 2007.

4. Floriberto Ortiz Rodriguez, Wen Yu, Marco A. Moreno-Armendariz, Nonlinear systems identification via two types of recurrent fuzzy CMAC, **2007 International Joint Conference on Neural Networks**, IJCNN'07, Orlando, USA, 2007.

5. Floriberto Ortiz Rodriguez, Wen Yu, Marco A. Moreno-Armendariz, Anti-swing control with hierarchical fuzzy CMAC compensation for an overhead crane, **2007 The 22nd IEEE International Symposium on Intelligent Control**, ISIC'07, Singapure, 2007.

6. Floriberto Ortiz Rodriguez, Wen Yu and Marco A. Moreno-Armendariz, Stable adaptive control with hierarchical fuzzy CMAC, **Fifth International Symposium on Neural Networks**, ISNN 2008, Beijing, China, September 24-28, 2008.

Capítulo 2

Red neuronal FCMAC

2.1. Sistema biológico del cerebro

2.1.1. Estructura del cerebro

El cerebro contiene muchos subsistemas especializados para realizar diferentes tareas que el cuerpo humano le exiga, las interacciones entre estos sistemas determinan su comportamiento, un esquema general de la anatomia del cerebro se muestra en la figura 2.1.

La figura 2.2 muestra las principales estructuras involucradas en el control motriz del cuerpo y sus conexiones internas, el sistema motriz funciona de una manera jerárquica, los niveles más altos de la jerárquia transmiten señales a los niveles bajos que procesan la información captada por los sensores y transmiten dicha informacion sobre su estado a los niveles altos.

La espina dorsal es la principal interfaz entre el cerebro y el resto del cuerpo, contiene muchos grupos de nervios sensoriales, motores (de ejecución) y muchos subsistemas neuronales que proporcionan los diferentes reflejos del cuerpo.

La base del cerebro esta compuesta de médula y la parte media del cerebro, contiene distintos grupos de neuronas especializadas para realizar diferentes tareas como pueden ser: mantenerse erguido, control del balance y transformación de la información sensorial, etc.

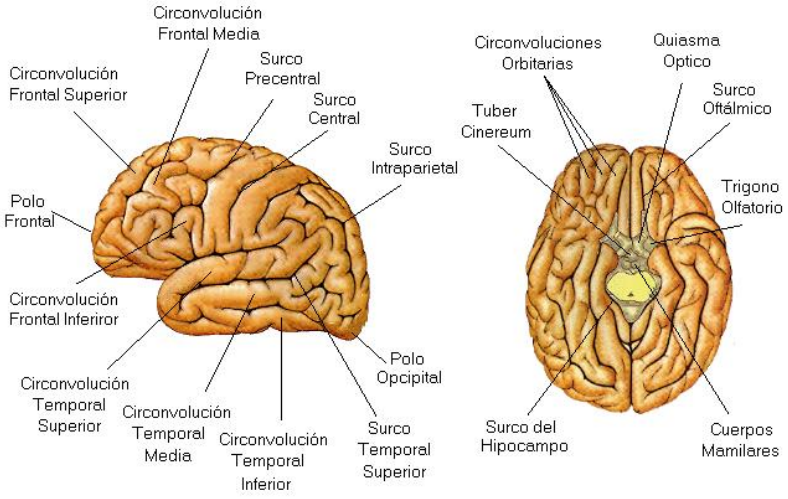

Figura 2.1: Estructura del Cerebro

El cerebelo ayuda en el aprendizaje y en coordinar los reflejos motrices aprendidos e instintivos, recibe información de la corteza matriz y de la espina dorsal, los cuáles son empledos con otras áreas del cerebro.

El tálamo procesa y distribuye la mayoría de la información motriz captada por los sensores y que es enviada a la corteza cerebral. Los ganglios basales estan al lado de la corteza cerebral y se encargan de coordinar algunos movimientos y estan involucrados en la parte cognocitiva del ser humano, pueden realizar las transiciones entre diferentes estados de movimiento.

La corteza cerebral, es un conjunto de neuronas finas que cubren al cerebro, el cuál presenta dos hemisferios: izquierdo y derecho, la corteza cerebral realiza funciones perceptivas, cognoscitivas, y las de más alto nivel motriz, así como la emoción y la memoria. Las respuestas de complejas comportamientos se originan aquí. Tiene muchas áreas que se especializan en diversas funciones cognoscitivas, tales como la visión, lengua, y planeamiento [32].

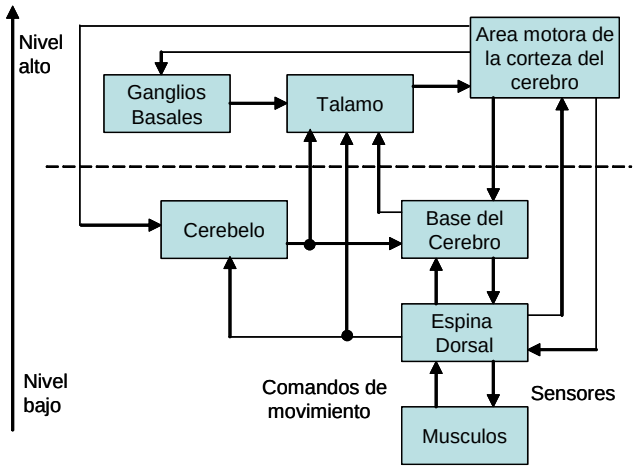

Figura 2.2: Esquema del control del movimiento humano

2.1.2. Neurona biológica

Una neurona biológica es una célula especializada que se encarga de transmitir información a otras células ó neuronas a partir de señales electro-químicas, como se observa en la figura 2.3, las señales eléctricas se presentan como diferencias de voltajes que atraviesan la membrana celular. Las neuronas presentan muchas ramificaciones que le permiten conectarse a otras neuronas, las cuáles se conocen con el nombre de dendritas, las dendritas (entradas) obtienen información de neuronas vecinas mientras que los axones transmiten (salidas) dicha información. Para distancias cortas entre neuronas la información es intercambiada a través de voltaje estático, para distancias grandes las señales son transmitidas a través de las dendritas y los axónes como una cadena de pulsos de voltaje, la información es codificada en una taza de pulso o de disparo.

Cada pulso transmitido a una neurona es integrado al cuerpo de la célula hasta alcanzar un umbral, con lo cual la neurona emite su propio pulso y la integración se reinicializa. La interfaz entre cualesquiera dos neuronas se conoce como sinapsis, la cuál se realiza entre un axón y una dendrita. La terminal del axón puede ser excitada o inhibida dependiendo del umbral, lo que significa que la sinapsis modula el pulso entrante de modo que tenga un

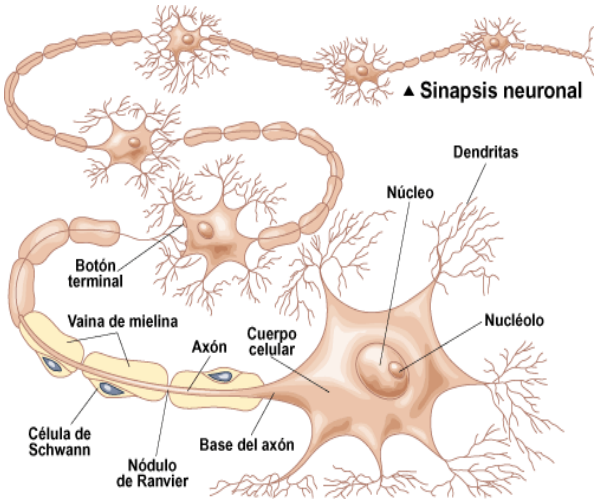

Figura 2.3: Neurona biológica

mayor o menor efecto de integración sobre la neurona final, este incremento o decremento es la taza de disparo.

Existen diferentes tipos de neuronas en el cerebro que presentan distintas formas y tamaños, por ejemplo las células granulares que están presentes en la corteza del cerebro son pequeñas con forma de estrella y solamente presentan en promedio cuatro dendritas (entradas), en contraste con las células Purkinje que son mucho más grandes, tienen una estructura en forma de arbol con muchas ramificaciones y pueden llegar a presentar cerca de 200,000 dendritas, como se observan en las figuras 2.4 y 2.5.

2.1.3. Aprendizaje

El aprendizaje se puede entender como un proceso neuronal en el cuál se realiza una interpretación interna del mundo externo, las diferentes estructuras del cerebro presentan diferentes comportamientos de aprendizaje utilizando varios mecanismos, en muchos casos el aprendizaje se da en los niveles de las neuronas individuales al ajustar su sinapsis ó alteraciones en su dinámica en respuesta a diferentes señales de entrada.

Figura 2.4: Células Granulares y Purkinge

En 1949 Donald Hebb propuso un mecanismo celular para el condicionamiento clásico conocido como aprendizaje hebbiano, figura 2.6. Un estimulo no condicionado es originado en la neurona A, esto estimula una respuesta en la neurona C porque se produce una sinapsis de acoplamiento entre las dos. Un estímulo condicionado que se origina en la neurona B, inicialmente no estimula una respuesta sináptica sobre C por ser pequeña, sin embargo, si B es estimulado al mismo tiempo que C es excitado, entonces la sinapsis $B - C$ será muy fuerte. Inicialmente C sólo es excitado en respuesta a A, si B estimula junto con A, entonces la sinapsis de B será muy grande y podrá influir en la respuesta de C, de esta manera un estimulo condicionado está asociado con un estimulo no condicionado.

2.1.4. Anatomía de la corteza del cerebelo

La tarea del cerebelo es regular la actividad motora que ocurre en otras áreas del cerebro, El cerebelo resulta especialmente vital para el control de actividades musculares rápidas, como correr, escribir a máquina, tocar el piano, incluso hablar, ver la figura 2.7. Las señales que se originan en el cerebelo modulan la cantidad del movimiento, el inicio y la terminación del movimiento y el control preciso del tiempo de cualquier evento que implique una secuencia coordinada de movimientos, regula y coordina los movimientos del cuerpo: Piernas, ojos,

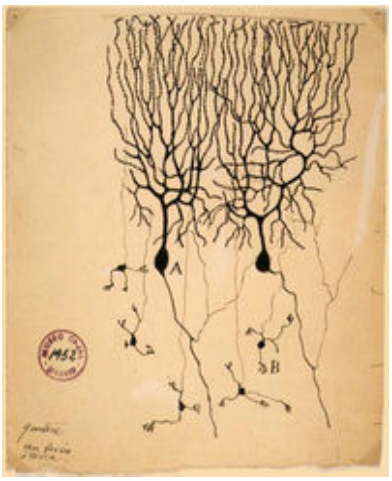

Figura 2.5: Células Purkinge

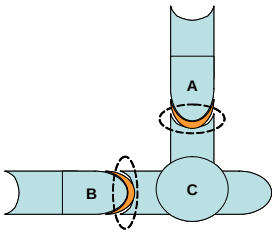

Figura 2.6: Aprendizaje Hebbiano

brazos, controlar los movimientos posturales y de equilibrio.

La pérdida de esta zona del encéfalo puede destruir cada una de estas funciones motoras. El cerebelo ocupa cerca del 10 % del volumen del cerebro, pero contiene más de la mitad del total de neuronas cerebrales, el cerebelo actúa en el control motriz sólo en relación con las actividades motoras que se inician en alguna otra parte del sistema nervioso. Pueden originarse en la médula espinal, la formación reticular, los ganglios basales o en áreas motoras de la corteza cerebral.

El cerebelo a pesar que no tiene el control directo sobre la contracción muscular vigila y establece ajustes correctores de las actividades motoras desencadenadas por otras partes

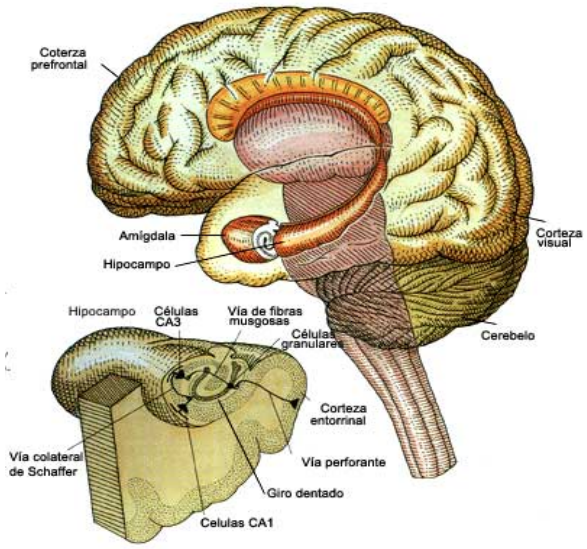

Figura 2.7: Anatomia del Cerebro

del encéfalo. Recibe continuamente información actual de las partes periféricas del cuerpo, para determinar el estado instantáneo de cada uno de sus áreas - su posición, su ritmo de movimiento, las fuerzas que actúan sobre él, etc. El cerebelo compara el estado físico actual de cada parte del cuerpo según indica la información sensorial, con el estado que intenta producir el sistema motor. Si los dos no se comparan favorablemente, de manera instantánea se trasmiten señales correctoras adecuadas hacia el sistema motriz, para aumentar o disminuir la actividad de músculos específicos. Los impulsos motores del cerebelo son transmitidos hacia los centros motores del cerebro y de la médula con destino a los músculos.

La corteza del cerebelo presenta tres capas de neuronas como se observa en la figura 2.8, las cuáles presentan diferentes tipos de células como son: células gránulosas, células purkinje, capa molecular, materia blanca (sólo axones). La corteza motora transmite señales a la periferia para causar una función motriz, pero al mismo tiempo transmite esta información al cerebelo, entonces el cerebelo compara las "intenciones"de la corteza con la .ªctuación"de las partes corporales, en caso que ésta no corresponda con aquéllas, calcula el .error.entre

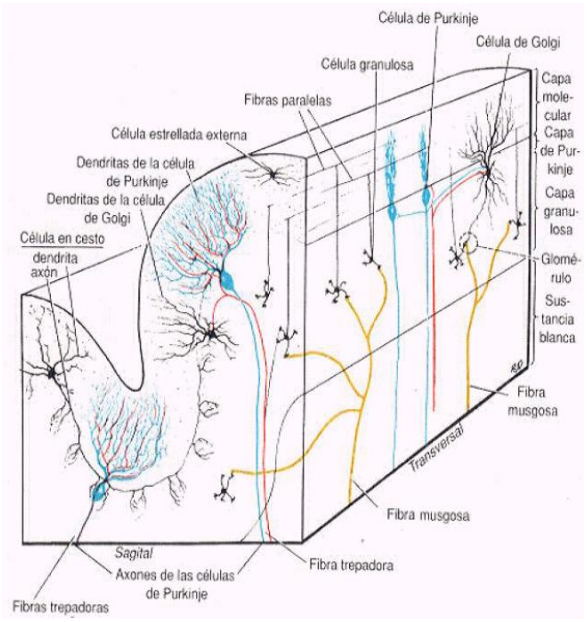

Figura 2.8: Estructura interna del cerebelo

ambas para poder llevar a cabo las correcciones apropiadas de inmediato. Un efecto secundario del mecanismo del cerebelo de retroalimentación es su capacidad de .ªmortiguar"los movimientos musculares. Para explicar el término .ªmortiguador", debemos señalar primero que prácticamente todos los movimientos del cuerpo "son pendulares". Debido a la inercia, todos los movimientos pendulares tienen tendencia a pasar a su estado inicial, si el cerebelo está intacto, señales subconscientes apropiadas detienen el movimiento exactamente en el sitio requerido, evitando así que se pase de él y suprimiendo el temblor de amortiguación esta es la característica básica de un sistema de amortiguamiento [31].

2.2. Redes neuronales artificiales

Las redes neuronales artificiales son un modelo simplificado de las redes neuronales biológicas, se pueden clasificar dentro del grupo de sistemas inteligentes, entre los que se

encuentran: sistemas adaptables, difusos, genéticos y todos aquellos que tratan de modelar el conocimiento y el aprendizaje. La calificación de inteligentes se debe a que los sistemas mencionados anteriormente tienen la capacidad de adaptarse a su medio ó aprender de él de forma autónoma. Las redes neuronales artificiales pueden definirse como un arreglo de elementos básicos de procesamiento con capacidad de entrenamiento. Este entrenamiento consiste en el ajuste de algunos parámetros con el fin de que la red asimile, con algún grado de precisión, la relación causa-efecto deseada entre las variables de entrada y de salida de la red neuronal. Las redes neuronales artificiales se han utilizado principalmente en dos grandes áreas:

- Aproximación de funciones
- Reconocimiento y clasificación de patrones

En el campo del control automático, las redes neuronales artificiales se utilizan principalmente como modelos para la identificación y el control de sistemas. La identificación de sistemas consiste en ajustar los parámetros de un modelo propuesto, de tal forma que su comportamiento se aproxime al sistema o planta a ser estudiado. La tarea de la identificación de sistemas es obtener modelos matemáticos que reproduzcan la relación entre las variables que intervienen en el sistema estudiado, las redes neuronales artificiales tienen una aplicación directa en el campo de identificación de sistemas debido a su capacidad de generalización a partir de la relación entrada-salida del sistema. La generalización de la red neuronal se efectúa aplicando un conjunto de entradas diferentes a las usadas en la etapa de aprendizaje y se observa su respuesta, esperando que reproduzca con cierto grado de precisión la respuesta del sistema identificado. Por otro lado, el problema de control busca diseñar un sistema que genere la señal de entrada a la planta necesaria para modificar su comportamiento con el fin de que se comporte de la forma deseada.

Dos tipos de redes neuronales han recibido mucha atención en trabajos recientes: Las redes neuronales multicapa y las redes neuronales recurrentes. Las redes multicapa han resultado una herramienta muy poderosa en problemas de reconocimiento de patrones (sistemas estáticos), mientras que las redes recurrentes han sido utilizadas para la identificación y control de

sistemas dinámicos, también en la resolución de problemas de optimización. Desde el punto de vista teórico las redes multicapas representan mapeos no lineales estáticos, mientras que las redes recurrentes son representados por sistemas con retroalimentación dinámica no lineal.

Se conoce con el nombre de redes neuronales artificiales a todos los modelos matemáticos que intenta imitar las capacidades y características de sus semejantes biológicos. Las redes neuronales están formados por elementos simples de cálculo, todos ellos interconectados con una cierta topología ó estructura, como lo son las neuronas llamadas perceptrones (Rossenblatt, 1958) que son el elemento más simple de una red. El modelo básico de una neurona está formada como se observa en la figura 2.9 por los siguientes elementos:

- Un conjunto de sinapsis, que son las entradas de la neurona ponderadas por un peso. Se identifica al peso que conecta la entrada s_j de la neurona i por el símbolo w_{ij}. Se incluye un peso independiente w_{i0} mejor conocido como bias, donde se considera una entrada constante para la neurona i.

- Un sumador que simula el cuerpo de la neurona y obtiene el nivel de excitación.

- La función de activación genera la salida si se alcanza el nivel de excitación y restringe el nivel de salida, evitando asi la saturación de la red.

La salida de la neurona está dada por la expresión:

$$y_i = \varphi\left(\sum_{j=1}^{n} w_{ij}s_j + w_{i0}\right) \tag{2.1}$$

en donde n indica el número de entradas a la neurona i.

El argumento de la función de activación es la combinación lineal de las entradas de la neurona. Si se considera al conjunto de entradas y los pesos de la neurona i como un vector de dimension $(n+1)$, la expresión (2.1) puede escribirse de la siguiente forma:

$$y_i = \varphi\left[W_i^T s\right] \tag{2.2}$$

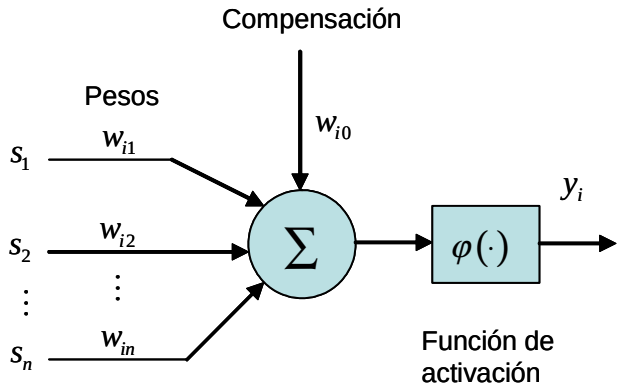

Figura 2.9: Elementos de una neurona.

donde $s = [-1, s_1, s_2, ..., s_n]^T$, $w_i = [w_{i0}, w_{i1}, ..., w_{in}]^T$. La neurona artificial descrita anteriormente está formado por un grupo de constantes que ponderan la entrada a un sumador y la salida es acotada por la función de activación. Las funciones de activación más utilizadas están definidas por las siguientes expresiones:

$$
\begin{array}{cc}
a) \text{ Escalón:} & b) \text{ Sigmoide} \\
\varphi(\cdot) = \sigma(s) = \begin{cases} 1, \text{ si } s \geq 0 \\ 0, \text{ si } s < 0 \end{cases} & \varphi(\cdot) = sigm(s) = \frac{1}{1+e^{-s}} \\
c) \text{ Tangente} & d) \text{ Saturación} \\
\varphi(\cdot) = \tanh(s) = \frac{1-e^{-2s}}{1+e^{-2s}} & \varphi(\cdot) = sat(s) = \begin{cases} -1, \text{ si } s < -1 \\ x, \text{ si } |s| \leq 1 \\ 1, \text{ si } s > 1 \end{cases}
\end{array}
\tag{2.3}
$$

Una propiedad de las funciones sigmoide y tangente hiperbólica es que su derivada existe y es una función continua. Esto es necesario ya que en el algoritmo de aprendizaje el cálculo del gradiente local para cada neurona del perceptrón multicapa requiere del conocimiento de la derivada de la función de activación $\varphi(\cdot)$ asociada con la neurona en cuestión. Para que esa derivada exista es necesario que $\varphi(\cdot)$ sea diferenciable. Las funciónes de activación

comúnmente utilizadas en redes neuronales son las funciones sigmoidal y gaussiana.

Las redes neuronales artificiales están construidas en base a elementos de cálculo relativamente simples, que son capaces de aprender de su ambiente y modificar su forma de interactuar con él. Al modo en que las neuronas se organizan en la red se le conoce como topología. En las redes neuronales artificiales se clasifican en tres arquitecturas elementales: redes con una sola capa, múlti-capas y en malla, como se observa en la figura 2.10.

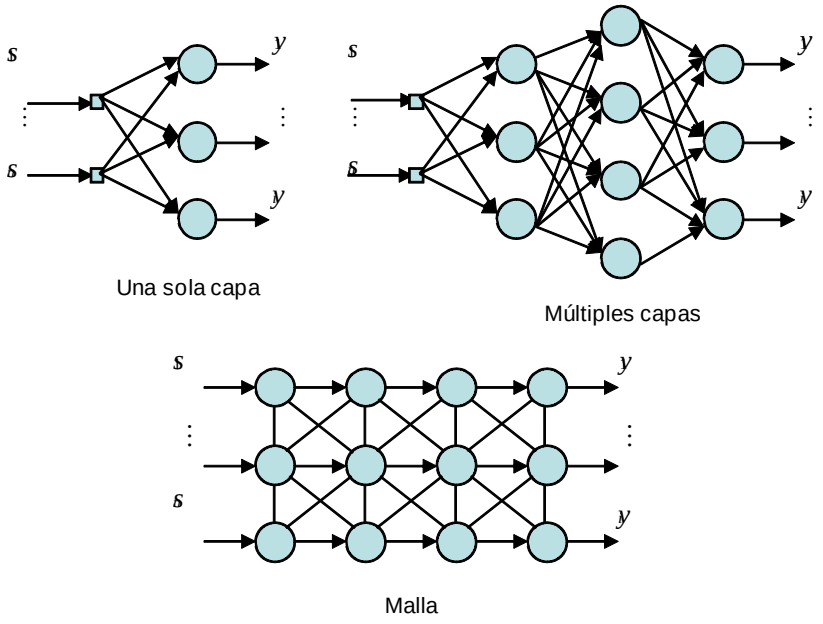

Figura 2.10: Topologia de las redes neuronales artificiales.

En las redes neuronales, la arquitectura usada influye en su capacidad de adaptación, debido al conjunto de pesos ajustables que presenta. Las redes neuronales presentan las siguientes características según su organización interna:

- Paralelismo
- Representación y cálculo distribuido
- Algoritmo de aprendizaje

- Tolerancia a fallas

Otra clasificación que presentan las redes neuronales se base en la manera en que la información se transmite en la red. Se dividen en dos grupos: redes con conexiones hacia adelante (estáticas) y redes con conexiones hacia atrás ó recurrentes (dinámicas).

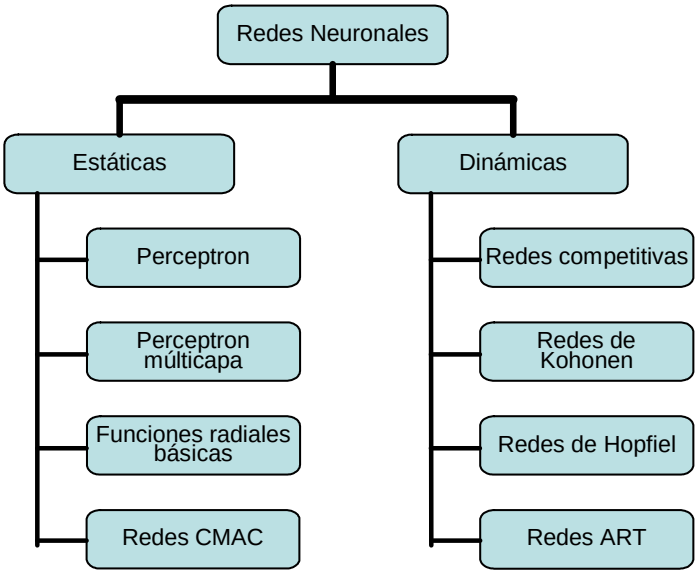

Figura 2.11: Redes estáticas y dinámicas

2.2.1. Redes neuronales estáticas

Si la respuesta de una red neuronal depende únicamente de sus entradas y no depende de señales en instantes anteriores de tiempo se dice que es una red neuronal estática, por lo que la respuesta de la red neuronal es invariante en el tiempo. Existen dos tipos de redes neuronales estáticas: las redes neuronales múlti-capa y las redes neuronales de funciones radiales básicas.

Las redes neuronales estáticas son muy útiles en los problemas de clasificación de patrones y aproximación de funciones porque construyen funciones no lineales entre el espacio de

entrada al espacio de salida de las variables involucradas. Una red neuronal con conexiones hacia adelante con una capa oculta no lineal y una capa de salida lineal puede aproximar cualquier función con el grado de precisión que se desee, como se demostró en: [14], [18], [17].

Redes neuronales múlticapas

Estas redes neuronales están construidas por neuronas organizadas en capas. Cada nivel intermedio de la red tiene como entrada a todas ó a un conjunto de las salidas de la capa anterior. Un ejemplo de una red neuronal con múltiples capas se presenta en la figura 2.10. La primer capa es la entrada a la red, su función es la distribuir las entradas de la red en la primer capa oculta, la capa dos está localizada entre la primer capa y la última capa, se denomina capa oculta, la última capa genera las salidas de la red neuronal. El flujo de la información se transmite en un sólo sentido, de manera que una red neuronal múlticapa es un mapeo no lineal del espacio de entrada s, al espacio de salida y, es decir: $s \rightarrow y$.

Redes neuronales con funciones de base radial

Un incoveniente de las redes neuronales MLP es que su entrenamiento es lento, la minimización del índice del error cuadrático de salida requiere de comparar en varias ocasiones el conjunto de datos de entrenamiento con la respuesta de la red neuronal. Las redes neuronales con funciones de base radial (RBF) son una alternativa a las redes neuronales MLP, en el contexto de que las RBF las capas ocultas están conformadas por un conjunto de funciones que constituyen una base para el problema de clasificación. La justificación matemática la establece el teorema de Cover (Cover, 1965), se basa en que un problema de clasificación es más probable que sea linealmente separable si se transforma en otro de dimensión mayor. Las funciones de base radial fueron introducidas primero para la solución de problemas de interpolación multivariable. El trabajo pionero en este tema fue Powell, 1985. Broomhead y Lowe en 1988, exploraron por primera vez el uso de las redes neuronales con funciones de base radial para poder realizar una clasificción no lineal. A diferencia de la disposición que se tiene en las funciones de activación de la red MLP que permite construir modelos de

entrenamiento mediante el algoritmo de retro-propagación, las nuevas redes con funciones de base radial construyen sus modelos con funciones de activación que son diferente tanto en la capa oculta como en la capa de salida, esto es, una red RBF está diseñada con neuronas en la capa oculta activadas mediante funciones radiales de carácter no lineal con sus centros propios y en la capa de salida mediante funciones lineales. La estructura de las redes de base radial presenta tres capas bien definidas:

- La capa de nodos de entrada, completamente conectadas a las neuronas de la capa oculta.
- La capa oculta de neuronas que proveen una transformación no lineal activada por las funciones de base radial.
- La capa de salida, también completamente interconectada a la capa oculta y activada a través de una función lineal continua.

La construcción de una red RBF requiere de una mayor cantidad de neuronas en los nodos ocultos que en las redes MLP. Aunque las redes RBF no son comúnmente utilizadas en aplicaciones que impliquen un alto volumen de patrones de entrenamiento, se le reconoce como una red con una alta eficiencia en la fase de entrenamiento. El entrenamiento a diferencia de la red MLP usando el algoritmo de aprendizaje de retro-propagación, es solamente hacia adelante, de este modo la salida de una red RBF en general, está influenciada por una transformación no lineal originada en la capa oculta a través de la función de base radial y una lineal en la capa de salida a través de la función lineal continúa.

En la figura 2.12 se presenta una red RBF, donde s_i, son las entradas a la red $i = 1, 2, ..., n$; la salida está dada por $y = w\mu(s)$, en donde las μ_i son las funciones de base radial, para este caso son funciones gaussianas. Si la red neuronal presenta v neuronas en la capa oculta, entonces la salida de la red neuronal de base radial se expresa de la forma:

$$y = \sum_{i=1}^{v} w_i \mu_i(s) \tag{2.4}$$

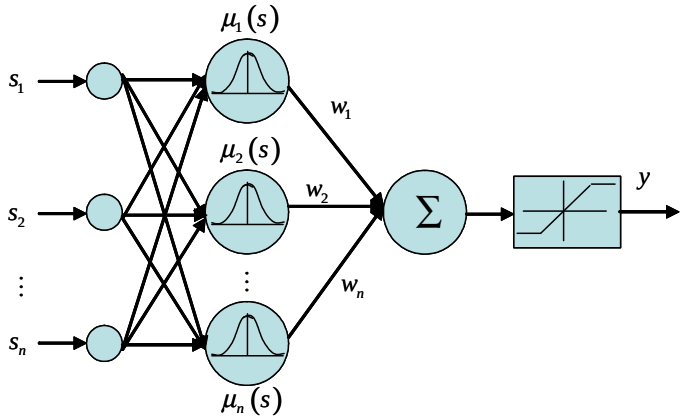

Figura 2.12: Red neuronal de funciones de base radial.

En general una red RBF tiene un mejor desempeño con un mayor volumen de datos de entrenamiento que su contraparte la red MLP, presentan una arquitectura simplificada con una capa oculta, su entrenamiento es rápido y se puede realizar una combinación de diferentes paradigmas de aprendizaje.

Aplicaciones de las redes estáticas

Las principales aplicaciones de las redes neuronales estáticas son aquellas en donde es necesario construir una relación entre la entrada y la salida, como por ejemplo:

- Clasificación de patrones: Sea d una muestra que pertenece a un conjunto D y c es un designador de clase. Se dice que d es de clase c, si existe una función f, tal que:

$$f : d \rightarrow c \tag{2.5}$$

  A todas las muestras m que cumplen 2.5, forman la clase C:

$$C = \{d \in D \mid f : d \rightarrow c\}$$

- Aproximación de funciones: Dada una función $f : s \rightarrow y$, en donde $s \in S \subseteq R^n$, $y \in Y \subseteq R^m$. Encontrar una $\hat{f}$ tal que:

$$\left\| f(s) - \hat{f}(s) \right\| < \epsilon; \quad \forall s \in S$$

  En donde ϵ es un entero positivo pequeño. Se dice que $\hat{f}$ es una aproximación en S de f.

- Memoria asociativa: En la actualidad las computadoras tienen memorias direccionadas, memorias que necesitan como información de entrada una dirección para recuperar un dato, esto hace que tengan una estructura rígida; en consecuencia tienen un tiempo de escritura y lectura grande. Por el contrario una memoria asociativa, no requiere de la dirección específica del dato guardado, lo que necesita es otro dato, es decir, en este tipo de memorias se recupera la información por la asociación impuesta sobre su contenido, este modo de operar hace a una memoria asociativa más rápida de recuperar la información.

Redes neuronales dinámicas

Hay problemas que necesitan de un sistema que tenga una respuesta dependiente de su estado anterior, por lo que requieren de una estructura que tenga una dinámica interna propia. Si la respuesta de una red neuronal depende de su pasado, se dice que es una red neuronal dinámica. Algunas de las tareas para este tipo de redes son: La predicción de series, la identificación y el control de sistemas dinámicos.

La figura 2.13 muestra las diferentes arquitecturas de las redes dinámicas. Las figuras (a) y (b) muestran redes con recurrencia local, la primera una recurrencia en la misma capa y la segunda muestra una recurrencia entre las capas ocultas. Las figuras (c) y (d) muestran una recurrencia global, donde la retroalimentación va de la salida de la red a la entrada y en el segundo caso la retroalimentación solo llega a la segunda capa de la red. En el desarrollo de este trabajo de tesis, las arquitecturas (a) y (c) fueron empleadas.

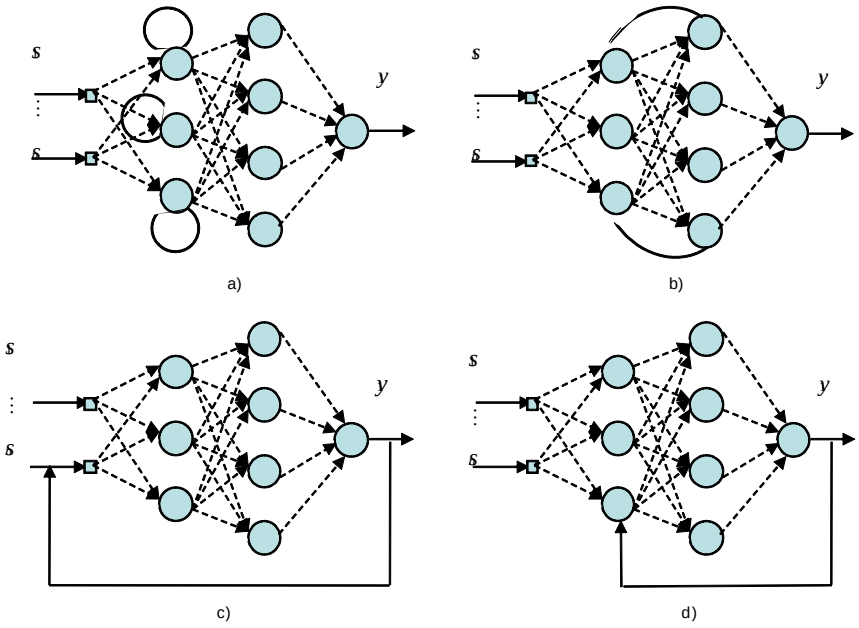

Figura 2.13: Arquitecturas de redes neuronales dinámicas

2.2.2. Aprendizaje en las redes neuronales artificiales

El aprendizaje en el contexto de las redes neuronales es un proceso por el cuál los pesos de la red neuronal son adaptados a través de un proceso de entrenamiento con datos del ambiente en el cuál la red se desea utilizar, el tipo de aprendizaje está determinado por la regla que se emplea para ajustar el valor de los pesos. Esta definición del proceso de aprendizaje implica la siguiente secuencia de pasos:

1. Las redes neuronales son estimuladas por el medio ambiente.

2. Las redes neuronales sufren cambios en sus pesos como resultado de esta estimulación.

3. Las redes neuronales responden de una nueva forma al ambiente por que los cambios han ocurrido en su estructura interna.

Al conjunto de reglas bien definidas para la solución del proceso de aprendizaje se le denomina algoritmo de aprendizaje. Los algoritmos de aprendizaje difieren unos de otros en la manera en la cuál ajustan los pesos de cada neurona.

Aprendizaje por corrección del error

La señal de salida de la neurona i es denotada por $y_i(k)$, esta señal de salida representa solamente la salida de la red neuronal, la cuál es comparada con la respuesta deseada, denotada por $d_i(k)$, consecuentemente se produce una señal de error denotada por $e_i(k)$, de esta definición se tiene:

$$e_i(k) = d_i(k) - y_i(k) \tag{2.6}$$

El algoritmo de aprendizaje se basa en el valor de la señal de error $e_i(k)$ para aplicar un ajuste en los pesos de la neurona i. En cada paso k los ajustes de corrección son realizados para hacer el valor de la señal de salida $y_i(k)$ muy cercano a la señal de la respuesta deseada $d_i(k)$. Este procedimiento es llevado a cabo al minimizar una función de costo ó indice de desempeño $\xi(k)$, definido en términos de la señal de error $e_i(k)$ como:

$$\xi(k) = \frac{1}{2}e_i^2(k) \tag{2.7}$$

Esto es, $\xi(k)$ es el valor instantáneo del error, el ajuste paso a paso de los pesos sinápticos de la neurona i son continuos hasta que el sistema alcance el estado estacionario (i.e., cuando los pesos no cambien), en este punto el proceso de aprendizaje término. El proceso de aprendizaje descrito anteriormente se refiere obviamente al aprendizaje por corrección del error, en particular la minimización de la función de costo $\xi(k)$ nos lleva a que la regla de aprendizaje comúnmente llamada la $regla \quad delta$ ó $regla \quad Widrow - Hoff$.

Sea $w_{i,j}(k)$ denota el valor del peso sináptico $w_{i,j}$ de la neurona i excitada por el elemento $s_j(k)$ por un vector de señales de entrada $s(k)$ en el tiempo k. De acuerdo a la regla delta, el ajuste $\Delta w_{i,j}(k)$ aplicado a los pesos sinápticos $w_{i,j}$ en el tiempo k está definido como:

$$\Delta w_{i,j}(k) = \eta e_i(k) s_j(k) \tag{2.8}$$

Donde η es una constante positiva que determina la tasa de aprendizaje de un paso del proceso de aprendizaje al siguiente. Nos referiremos a η como el parámetro de la tasa de aprendizaje. Se asume que la señal de error se puede medir.

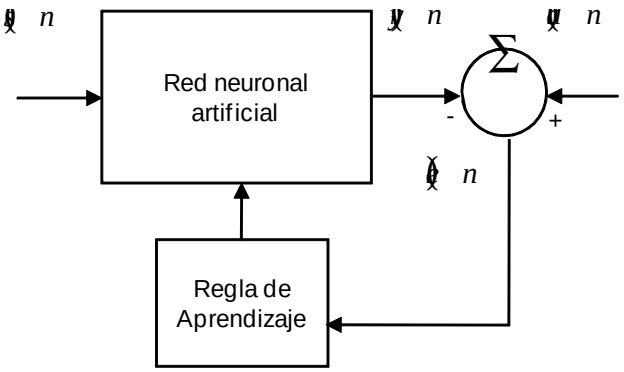

Figura 2.14: Diagrama a bloques de la red neuronal artificial.

De la figura (2.14) se observa que el aprendizaje por corrección del error es de naturaleza local, esto es, que los justes de los pesos sinápticos realizados por la regla delta están localizadas alrededor de la neurona i. Al calcular el ajuste sináptico $\Delta w_{i,j}(k)$, los valores actualizados de los pesos sinápticos $w_{i,j}$ están determinados por:

$$w_{i,j}(k+1) = w_{i,j}(k) + \Delta w_{i,j}(k)$$

$w_{i,j}(k)$ y $\Delta w_{i,j}(k)$ pueden ser vistos como el anterior y el nuevo valor de los pesos sinápticos $w_{i,j}$.

2.3. Sistemas difusos

La lógica difusa es una generalización de la lógica clásica, en la cuál existe una transición suave del valor falso al verdadero. La base de la lógica difusa es derivada de la teoría de

los conjuntos difusos [19], Un conjunto difuso A en S, está caracterizada por una función de pertenencia $\mu_A(s)$, el cuál asocia un número real con cada elemento en S en el intervalo $[0,1]$, con el valor de $\mu_A(s)$ se representa el grado de pertenencia de s en A. Esto es, el conjunto difuso A sobre el universo S se define como:

$$A = \{(s, \mu_A(s)) \mid s \in S\}$$

La función de pertenencia mapea cada elemento a su valor de pertenencia entre 0 y 1. Algunos tipos de funciones de pertenencia son: triangular, trapezoidal, gaussiana, campana generalizada, sigmoidal, etc.. A continuación se presenta brevemente los conceptos básicos de los sistemas difusos.

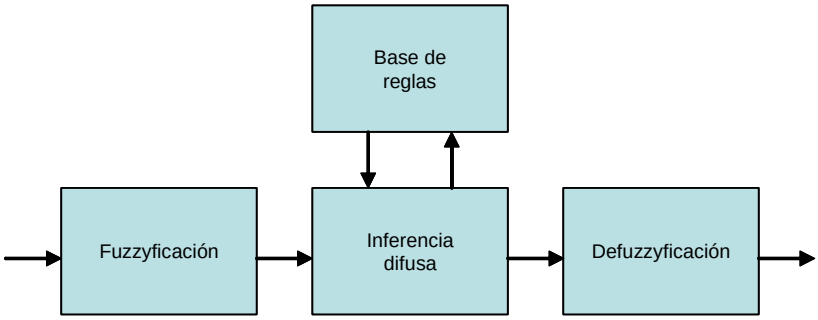

Figura 2.15: Componentes de un sistema difuso.

Un sistema difuso es un mapeo no lineal de un vector de datos de entrada a una salida escalar utilizando lógica difusa. Este mapeo se realiza utilizando la fuzificación, inferencia difusa, y la defuzificación, componentes que conforman un sistema difuso como se muestra en la figura 2.15. El componente de fuzzyficación mapea un valor no difuso del espacio de entrada a un valor lingüistico conveniente, el cuál puede ser visto como una etiqueta del conjunto difuso. La máquina de inferencia difusa consiste de tres partes conceptuales: La base de reglas que contiene una selección de las reglas difusas, la base de datos que definen las funciones de pertenencia utilizadas en las reglas difusas, la normalización de las entradas y salidas del universo del discurso, La realización de la partición difusa del espacio de entrada

y salida, y el mecanismo de razonamiento que realiza el procedimiento de inferencia sobre las reglas y da las condiciones para derivar una salida razonable. La estructura de una regla difusa sencilla tiene la estructura:

$$SI\ s \text{ es } A,\ ENTONCES\ y \text{ es } B \tag{2.9}$$

Donde A y B son los valores lingüísticos de entrada y salida definidos por los conjuntos difusos sobre el universo S y Y respectivamente. La parte $SI\ s$ es A, es llamada la parte antecedente de la regla, la parte $ENTONCES\ y$ es B, es llamada la parte consecuente. La regla puede ser representada como una relación de implicación como:

$$R : A \rightarrow B$$

Donde R es la relación difusa definida sobre el universo del producto cartesiano $S \times Y$. A y B son la parte antecedente y consecuente de la regla. Esta relación puede ser obtenida de diferentes fuentes: del conocimiento y experiencia de un experto, del sentido común, del conocimiento intuitivo del ingeniero de diseño, acerca del sistema bajo estudio ya sea de los principios físicos de funcionamiento o de las leyes físicas que lo gobiernan.

Los modelos difusos básicamente se dividen en dos categorias: Uno es el modelo lingüistico, el cuál está basado en una colección de reglas $si-entonces$, que utilizan un razonamiento difuso este se conoce como modelo Mamdani. La forma de este modelo para un sistema de dos entradas-una salida se describe por:

$$SI\ s_1 \text{ es } A \text{ y } s_2 \text{ es } B,\ ENTONCES\ y \text{ es } C \tag{2.10}$$

donde A, B y C son conjuntos difusos sobre el universo S, Y y Z respectivamente.

El otro tipo es el modelo Sugeno, está caracterizado en su parte consecuente por una expresión matemática:

$$SI\ s_1 \text{ es } A \text{ y } s_2 \text{ es } B,\ ENTONCES\ y = f\left(s, y\right) \tag{2.11}$$

donde A y B son conjuntos difusos sobre el universo del discurso S y Y respectivamente. Un modelo Sugeno de primer orden tiene la forma:

$$SI\ s_1 \text{ es } A \text{ y } s_2 \text{ es } B,\ ENTONCES\ y = ps + qy + r \tag{2.12}$$

Los modelos difusos pueden ser desarrollados empleando el método directo, donde se extrae directamente el conocimiento del experto y es expresado en la forma de reglas lógicas.

2.4. Red neuronal CMAC

Una de las técnicas de control inteligente más utilizadas y que ha presentado un incremento significativo en su investigación y desarrollo en los últimos años han sido las redes neuronales artificiales (NNs, por sus siglas en inglés), las cuáles son muy poderosas para identificar y controlar una amplia variedad de sistemas no lineales usando únicamente entradas y salidas de la planta sin necesidad de tener con un modelo matemático que la describa. Los controladores basados en redes neuronales han sido desarrollados para compensar los efectos no lineales presentes en los sistemas a controlar. En la topología de las redes neuronales hacia adelante (feedforward) todos los pesos son actualizados en cada ciclo de aprendizaje, este aprendizaje es de naturaleza global, lo cual es una tarea que consume mucho tiempo, esto llega a ser una desventaja cuando se pretende realizar un aprendizaje en linea o para aplicaciones de control en tiempo real donde el tiempo de respuesta se vuelve crítico.

Una nueva topología de red neuronal artificial que no presenta estas desventajas fue propuesta por James Albus en 1976 llamada: Cerebellar Model Articulatión Controller - CMAC -, cuya traducción literal es modelo del cerebelo para el control de la articulación [1], [2]. Este modelo artificial es una representación del modelo biológico que se describe de manera general por tres capas que corresponden a las neuronas sensoriales que detectan los estímulos, las células granulares y las células Purkinje, estás últimas se encuentran en la corteza del cerebelo [31]. La red neuronal CMAC puede ser clasificada como un perceptrón de memoria asociativa o mecanismo de búsqueda (look-up) por medio de una tabla, el cuál no se encuentra completamente conectada y la actualización de los pesos es local; ha sido utilizado ampliamente para control de sistemas en lazo cerrado de sistemas dinámicos complejos en tiempo real debido principalmente a su rápido proceso de aprendizaje. La red CMAC puede aprender relaciones no lineales de una amplia categoria de funciones y es una alternativa a las redes clásicas como el MLP. Algunas desventajas de las redes multicapa como el Perceptron

(MLP) es que su algoritmo de entrenamiento generalmente toman muchas iteraciones para converger, de aquí que en aplicaciones de aprendizaje en linea y en tiempo real no son muy recomendables, el número de cálculos por iteración es grande y el algoritmo se ejecuta lentamente, lo que hace muy difícil su implementación en hardware. La figura 2.16, muestra de manera general las diferentes topologías de redes CMAC que son estudiadas en este trabajo, en las siguientes secciones se explica a detalle el funcionamiento de cada una de ellas.

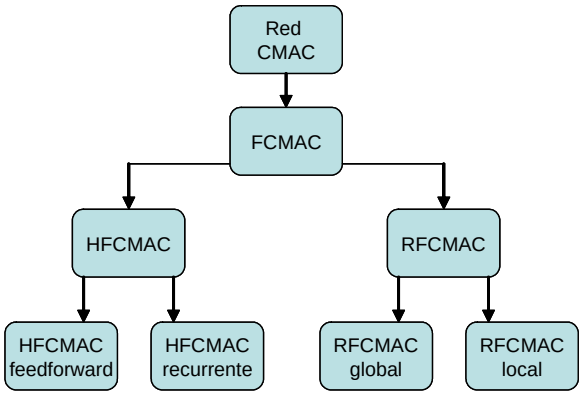

Figura 2.16: Diferentes topologias de redes CMAC.

2.4.1. Red neuronal CMAC

La idea básica de la red CMAC es almacenar los datos dentro de una región, de tal forma que los datos puedan fácilmente recuperarse y el espacio de almacenamiento sea mínimo, la red CMAC consta de múltiples entradas y múltiples salidas. Por las características mencionadas en la introducción, la red CMAC es considerada una red neuronal artificial, sin embargo, para las implementaciones actuales en software, generalmente es mucho más conveniente tratar a la red CMAC como una tabla de búsqueda. La red CMAC de la figura 2.17 consta de tres capas $L1, L2$ y $L3$.

La capa $L1$ presenta las variables de entrada $s = [s_1, s_2, ..., s_n]$, el espacio de entrada es

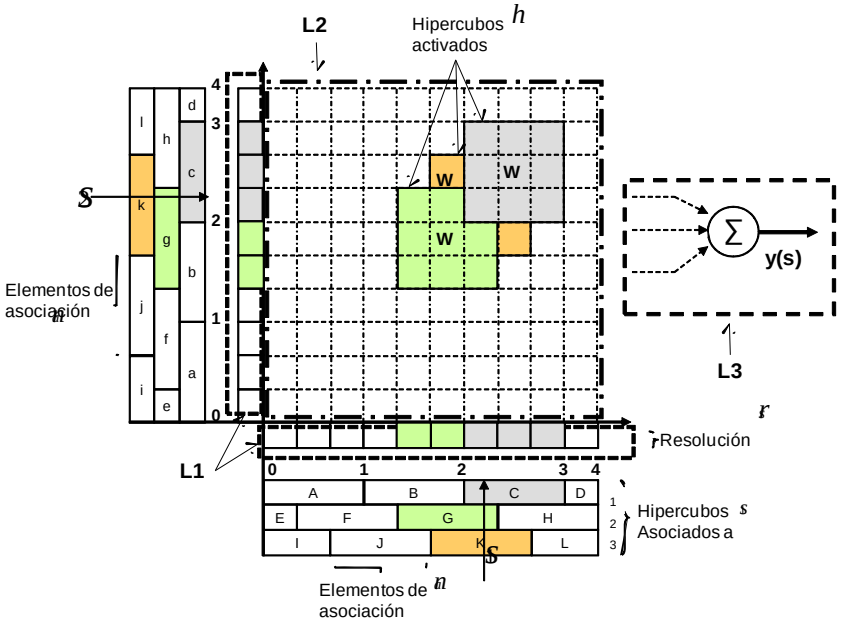

Figura 2.17: Estructura interna de la red CMAC.

dividido en segmentos denominados elementos de resolución denotado por r_s, cada espacio de entrada tiene un rango de valores permitidos, por lo que cada entrada analógica s_i se cuantiza y se convierte en un valor discreto, este nuevo par de valores será utilizado para generar una dirección de memoria.

En la capa $L2$, la dirección de memoria seleccionada previamente se asocia con otras memorias cercanas a su vecindad, este conjunto de memorias se denominan elementos de asociación n_a, al proyectar esta vecindad sobre los elementos de resolución r_s para cada entrada s_i, se forma el ancho de asociación. Para la misma dirección de memoria se le pueden asociar diferentes localidades de memoria, a cada grupo con diferentes localidades de memoria, se le conoce como hipercubo h. El contenido de todos los hipercubos que son memorias presentan un valor numérico que será actualizado mediante un algoritmo de aprendizaje.

La capa de salida $L3$ es la suma del contenido de todos los elementos de memoria de los hipercubos, la salida total de la red CMAC se representa por $y\,(s)$.

Los elementos de resolución r_s, el rango de los valores de entrada permitidos en la red CMAC, los elementos de asociación n_a, el número de hipercubos h y su tamaño, son definidos por el diseñador.

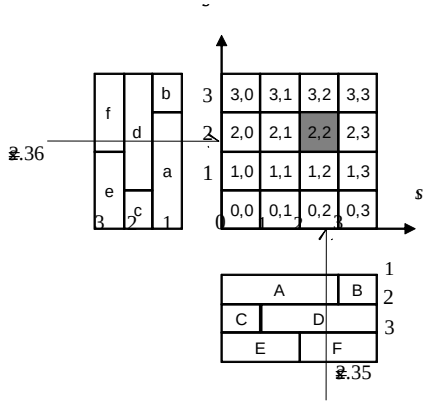

Figura 2.18: Red CMAC: dos entradas, una salida.

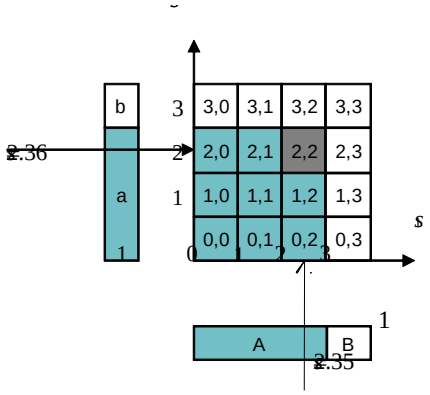

Figura 2.19: Elementos de Asociación 1.

Un ejemplo de la red CMAC con dos entradas-una salida se observa en la figura 2.18. En la capa $L1$ se encuentran las variables de entrada $s_1 = 2{,}35$ y $s_2 = 2{,}36$,. Los elementos de resolución son $r_s = 3$, el rango de valores permitidos es de 0 a 3 por cada entrada s_i,

el número de asociación es de 3, y el número de hipercubos es 3. Las entradas originales se cuantizan y toman nuevos valores, esto es: $s_1 = 2$ y $s_2 = 2$, este valor genera la dirección de memoria $(2, 2)$.

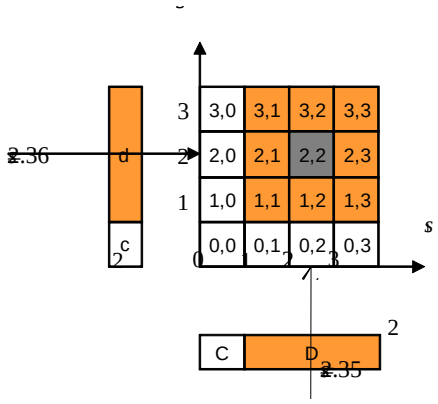

Figura 2.20: Elementos de asociación 2.

Como se observa en la figura 2.19, los primeros elementos de asociación que corresponden a la dirección $(2, 2)$ son las localidades: $(0, 0)$, $(0, 1)$, $(0, 2)$, $(1, 0)$, $(1, 1)$, $(1, 2)$, $(2, 1)$, $(2, 2)$. Todos estos elementos de memoria forman el primer hipercubo h_1.

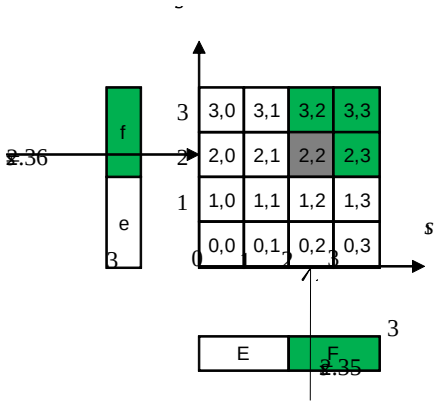

Figura 2.21: Elementos de asociación 3.

El segundo grupo de elementos de asociación que corresponden a la dirección $(2,2)$ son: $(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)$, este conjunto de elementos de memoria forman el hipercubo dos h_2, como se observa en la figura 2.20.

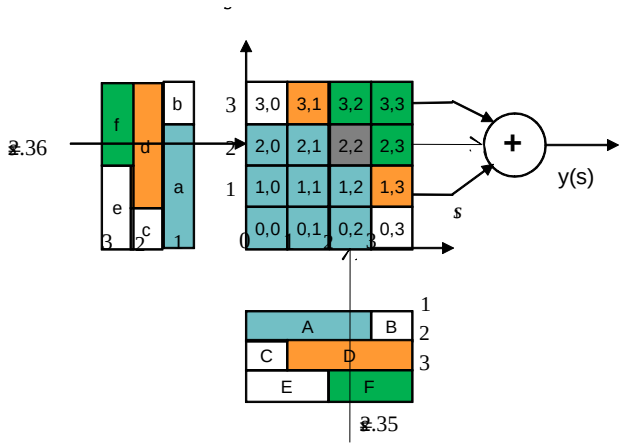

Figura 2.22: Total de elementos activados en la red CMAC.

Por último, el hipercubo $h3$ consta de los elementos de memoria: $(2,2),(2,3),(3,2),(3,3)$, como se muestra en la figura 2.21. El total de elementos de memoria activados que se muestran en la figura 2.22, son $(0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)$.

Al número total de combinaciones de memoria también se puede realizar mediante todas las combinaciones de hipercubos como se muestra en la figura 2.23, a todos los hipercubos activados por las entradas se les asigna el valor de *uno*, a las bloques de asociación no activados se les asigna el valor de *cero*, los hipercubos activados están compuestos de elementos de memoria ó neuronas, estos serán los únicos que podrán actualizar sus pesos mediante algún algoritmo de aprendizaje. El total de hipercubos ó combinaciones que se pueden formar son:

$$[Aa, Ab, Ba, Bb, Cc, Cd, Dc, Dd, Ee, Ef, Fe, Ff] \tag{2.13}$$

La tercera capa de la red, presenta la sumatoria de todos los pesos de las neuronas activadas en la capa $L2$, del total de neuronas presentes en la red, solo una pequeña cantidad

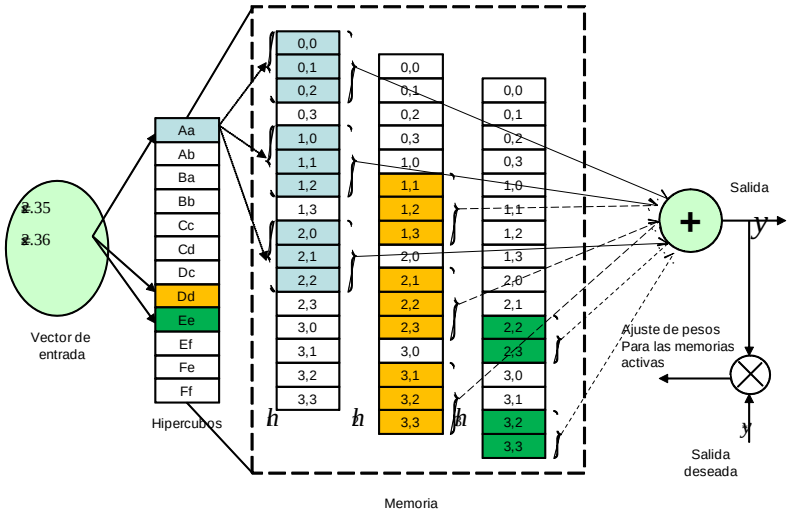

Figura 2.23: Hipercubos activados en la red CMAC.

de ellas son utilizadas. La salida de la red CMAC como se observa en la figura 2.23, es el valor numérico contenido en cada dirección de memoria física activada por los hipercubos, son sumadas para seleccionar un único conjunto de pesos, estos definen el valor de la función de control almacenada en esa dirección y se encuentra dada por:

$$\begin{aligned} y &= \begin{bmatrix} w(Aa) + w(Ab) + w(Ba) + w(Bb) + \\ w(Cc) + w(Cd) + w(Dc) + w(Dd) + \\ w(Ee) + w(Ef) + w(Fe) + w(Ff) \end{bmatrix} \\ y &= [w(Aa) + w(Dd) + w(Ff)] \\ y_i &= \mathbf{a}_i \mathbf{w} = \sum_{j=1}^{N_h} a_{ij} w_j \end{aligned} \tag{2.14}$$

Donde w es el vector columna con el contenido de las memorias activadas y a_{ij} es el vector fila con los elementos de la memoria activados. Cuando se tiene el caso de un espacio de múltiples entradas - múltiples salidas los requerimientos de memoria aumentan.

2.4.2. Red neuronal CMAC generalizada

La figura 2.24, muestra la arquitectura de una red CMAC generalizada estudiadas en [6], [11], a diferencia de la red CMAC propuesta por J. Albus es que esta nueva topologia de red presenta funciones gaussianas en los hipercubos.

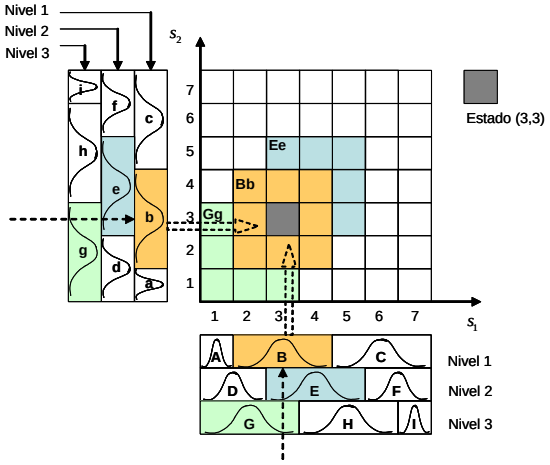

Figura 2.24: Red CMAC generalizada

La red CMAC generalizada está compuesta por cinco capas, la propagación de los datos de entrada y las funciones básicas en cada etapa se explican a continuación:

1. En la capa *uno* se encuentra el espacio de entrada s, el cuál es continúo y $n-$ *dimensional*, se define como: $s = [s_1, s_2, ..., s_n]^T \in R^n, i = 1, .., n$, donde n denota el número de entradas. Cada entrada s_i es dividida en varios elementos, el número de elementos se conoce como resolución r_s, algunos de estos elementos son agrupados para formar un bloque n_b por cada entrada, llamado bloque de asociación.

2. La capa *dos* es conocida como espacio de asociación de memoria A: El número total de bloques n_b formados en los diferentes niveles de quantización se define como n_A. El número de bloques n_A en la red CMAC generalmente es mayor que *dos*. La intersección de los bloques de asociación n_b se conoce como hypercubos h_b. El comportamiento de

cada bloque h_b está en función de los campos receptivos, que es la diferencia principal con la red CMAC original, para una red neuronal CMAC generalizada los campos receptivos pueden ser generalmente funciones triangulares ó gaussianas. En este trabajo se utiliza la función gaussiana representada como:

$$\mu_{A_j^i}(s_i) = \exp\left[\frac{-(s_i - c_{ij})^2}{\sigma_{ij}^2}\right], \quad j = 1, ..., n_A \tag{2.15}$$

Donde $\mu_{A_j^i}(s_i)$ representa la funcion gaussiana asociadas a cada campo receptivo j. El superíndice i representa la $i-ésima$ entrada s_i, con el promedio c_{ij} y la varianza σ_{ij} para cada una de las entradas.

3. Espacio del campo receptivo T: Las áreas formadas por los bloques como en el vector (2.13), son llamados campos receptivos asociados a una función gaussiana, el número de campos receptivos es igual a n_b (bloque de asociación). Cada localidad de A corresponde a un campo receptivo. El campo receptivo multidimensional se define como:

$$\begin{aligned} b_j(s, c_j, v_j) &= \prod_{i=1}^{n} \mu_{A_j^i}(s_i) \\ &= \exp\left[\sum_{i=1}^{n} \frac{-(s_i - c_{ij})^2}{\sigma_{ij}^2}\right], \quad j = 1, ..., n_r \end{aligned} \tag{2.16}$$

Donde b_j está asociado con el $j-ésimo$ campo receptivo, $c_j = [c_{1j}, c_{2j}, ..., c_{nj}]^T \in R^n$ y $v_j = [\sigma_{1j}, \sigma_{2j}, ..., \sigma_{nj}]^T \in R^n$. $i = 1, \cdots, n$. El campo receptivo multidimensional puede ser representado en una notación vectorial como:

$$\varphi(\mathbf{s}, \mathbf{c}, \mathbf{v}) = [b_1, b_2, ..., b_{n_r}]^T$$

Donde $\mathbf{c} = \left[\mathbf{c}_1^T, \mathbf{c}_2^T, ..., \mathbf{c}_k^T, ..., \mathbf{c}_{n_r}^T\right]^T \in R^{n \times n_r}$ y $\mathbf{v} = \left[\mathbf{v}_1^T, \mathbf{v}_2^T, ..., \mathbf{v}_k^T, ..., \mathbf{v}_{n_r}^T\right]^T \in R^{n \times n_r}$.

4. Espacio de memoria de los pesos $\mathbf{w}$: cada localidad de T presenta un valor ajustable en el espacio de memoria de pesos, con n_r componentes puede ser expresado como:

$$\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_o, ..., \mathbf{w}_p]$$

donde $\mathbf{w}_o = [w_{1o}, w_{2o}, ..., w_{jo}, ..., w_{n_r o}]^T \in R^{n_r}$, w_{jo} denota el valor de los pesos de la $o - ésima$ salida asociada con el $j - ésimo$ campo receptivo. Los pesos de w_{ko} son inicializadoas en cero y se ajustan automáticamente durante la operación en línea.

5. El espacio de salida y: La salida de la red CMAC es la suma algebraica de los valores de los campos receptivos $Bb, Ee, Gg, ..$ activados por las entradas, el cuál se expresa como:

$$\begin{aligned} y &= \sum w_{jo}\varphi_j(\mathbf{s}, \mathbf{c}_j, \mathbf{v}_j) \\ y &= w^T\varphi(s) \end{aligned} \tag{2.17}$$

2.4.3. Red neuronal CMAC difusa

La arquitectura de una red CMAC difusa se muestra en la figura 2.25, esta red está compuesta por cinco capas: En la primer capa se encuentra el espacio de entrada, en la segunda se encuentra la capa de *fuzzificación*, en la tercera está la capa de asociación difusa, en la cuarta la capa de post-asociación difusa y en la última, la capa de salida. Las funciones básicas realizadas en cada capa se explican a continuación:

1. La capa L_1 está dada por $s = [s_1, s_2, ..., s_n]^T \in R^n$, la cual transfiere las entradas de la red a la siguiente capa dada por la variable $mf_i = s_i$, $i = 1, ...n$, donde n es el número de las variables de entrada.

2. La capa L_2 de la red neuronal CMAC difusa es conocida como capa de *fuzzificación*, en donde cada nodo de *fuzzificación* corresponde a una variable lingüistica (ej. positivo, cero, negativo), la cuál esta expresada por una función de pertenencia $\mu_{A_j^i}$, en este trabajo se utiliza la función gaussiana representada como en 2.15. Existen m variables lingüisticas por cada entrada como se observa en la figura 2.25, el número de nodos en esta capa es n^m.

3. La capa L_3 de asociación difusa realiza el cálculo de la parte antecedente de las reglas lógicas difusas. Cada nodo en esta capa realiza la operación de implicación:

SI s_1 está en A_j^1 y s_n está en A_j^n ENTONCES ...

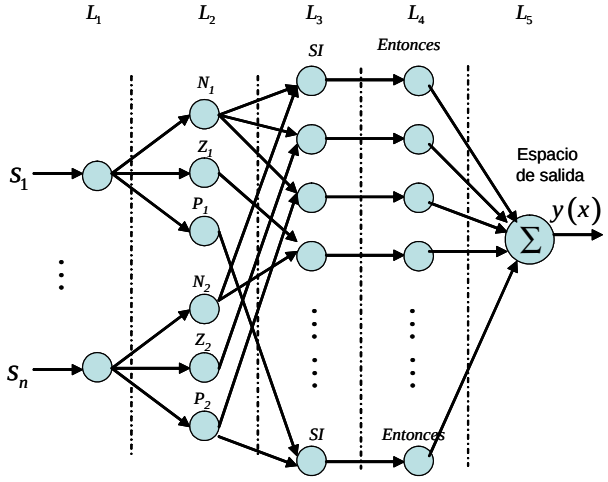

Figura 2.25: Red neuronal CMAC difusa

Se utiliza la regla producto y se obtiene:

$$\alpha_p = \prod_{i=1}^{n} \lambda_p \left(\mu_{A_j^i} \right) \tag{2.18}$$

donde p son las veces de asociación, $p = 1, ..., l$ donde l es el número de asociación, λ es el vector de selección de la memoria de asociación la cuál está definida como:

$$\lambda_p \left(\mu_{A_j^i} \right) = \mu_{A_{j,p}^i} = [0, 0 \cdots 1, 0 \cdots] \begin{bmatrix} \mu_{A_1^i}, \\ \vdots \\ \mu_{A_m^i} \end{bmatrix} \tag{2.19}$$

4. La capa L_4 de post-asociación difusa se calcula al normalizar las reglas activadas y prepararlas para la inferencia difusa.

$$\varphi_p = \alpha_p / \sum_{p=1}^{l} \alpha_p = \left(\prod_{i=1}^{N_i} \mu_{A_j^i,} \right) / \left(\sum_{j=1}^{N_A} \prod_{i=1}^{N_i} \mu_{A_j^i,} \right) \tag{2.20}$$

5. El espacio de salida L_5: La salida de la red CMAC difusa es la suma de los pesos activados por las entradas en el espacio de memoria, la cuál se expresa como:

$$y = w^T \varphi (s) \tag{2.21}$$

Red CMAC via sistemas difusos

Para un sistema difuso con N número de reglas difusas y *dos* variables de entrada $s = [s_1, s_2]$, se construyen las reglas difusas con la siguiente estructura:

$$R^i : \text{Si } \ s_1 \text{ está en } A_1^i \text{ y } \ s_2 \text{ está en } A_2^i, \text{ ENTONCES } y = w^T \varphi(s) \tag{2.22}$$

Donde $i = 1, 2, ..., N$ y la parte consecuente de la regla se obtiene de la salida de la red CMAC. Las funciones de pertenencia utilizadas para los conjuntos difusos A_j^i son gausianas $\mu_{A_j^i}$, donde $j = 1, .., m$. Se realiza el proceso de fuzzificación para calcular la salida y:

$$\begin{aligned} y &= \frac{v_1\varphi_1^T w + v_2\varphi_2^T w + ... + v_N\varphi_N^T w}{v_1 + v_2 + ... + v_N} \\ &= \frac{\sum_{i=1}^{N} a_i^T w v_i}{\sum_{i=1}^{N} v_i} \end{aligned} \tag{2.23}$$

Donde: $v_1 = \prod_{i=1}^{2} \mu_{A_j^i}(s_i)$, la ecuación anterior puede ser escrita en forma compacta como:

$$y = w^T \varphi(s) \tag{2.24}$$

Donde: $\varphi = [\varphi_1, \varphi_2, ..., \varphi_N]^T$, $\varphi_i = \frac{v_i}{\sum_{i=1}^{N} v_i}\Delta$, $\Delta = \left[a_1^T, a_2^T, ..., a_N^T\right]^T$

La matriz Δ se determina por los vectores de asociación de la red CMAC y φ se determina por las reglas difusas, las cuáles podrán ser dadas y permitirán al vector de pesos w ajustar sus valores [13].

Por simplicidad para el problema de *dos* dimensiones, se construye la base de reglas difusas con la estructura (2.22), las funciones de pertenencia se elijen como en la figura 2.26, en la cuál P es positivo y N es negativo son conjuntos difusos para cada variable, existen cuatro reglas con *cuatro* vectores de asociación a_1, a_2, a_3, a_4 denotados por las siguientes combinaciones $[s_1, s_2] = [(P, P), (P, N), (N, P), (N, N)]$. Para este ejemplo existen 16 vectores de asociación en la red CMAC, mientras que en le red CMAC difusa solo *cuatro* son utilizados, para determinar el vector de asociación a_i en la red CMAC difusa, se realiza la operación lógica

OR en todos los vectores de asociación de la red CMAC. Por ejemplo si $[s_1, s_2]$ está en el caso (P, P) existen nueve elementos en la región y al realizar la operación OR sobre los correspondientes nueve vectores de asociación se obtiene $a_1 = [1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1]$.

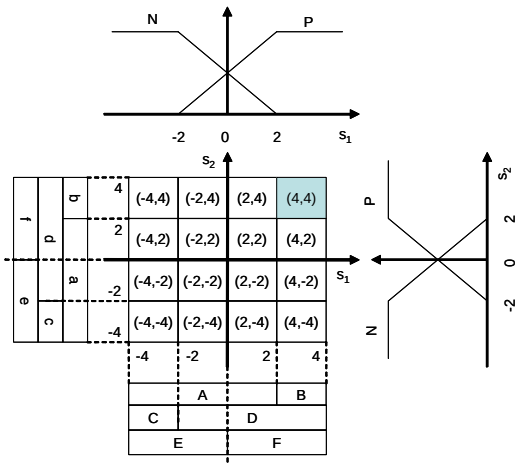

Figura 2.26: Relación CMAC con las reglas difusas

De la figura 2.26, se descompone en diferentes niveles de quantización, donde cada hipercubo generado tendrá asociado una función de pertenencía, en este caso es una función triangular, dentro de los hipercubos generados existen células o neuronas, analizaremos la célula ubicada en las coordenadas $(4, 4)$. Se construye la base de reglas difusas en base a las posibles combinaciones de las funciones de pertenencia como:

$$R^1 : SI\ s_1 = P\ y\ s_2 = P,\ ENTONCES\ y = w^T\varphi_1$$
$$R^2 : SI\ s_1 = P\ y\ s_2 = N,\ ENTONCES\ y = w^T\varphi_2$$
$$R^3 : SI\ s_1 = N\ y\ s_2 = P,\ ENTONCES\ y = w^T\varphi_3$$
$$R^4 : SI\ s_1 = N\ y\ s_2 = N,\ ENTONCES\ y = w^T\varphi_2$$

La parte antecedente de las reglas difusas corresponden al vector de selección de los elementos de memoria, para el ejemplo de la célula $(4, 4)$, en el primer nivel de quantización, dispara el hipercubo $Bb = 1$, en el segundo nivel dispara $Dd = 1$ y en el tercero $Ff = 1$, Estos son los elementos de memoria activados, se realiza el mismo procedimiento para cada una de

las células. La parte ENTONCES es la suma de todos los elementos de memoria activados en las diferentes combinaciones $((P,P),(P,N),(N,P),(N,N))$. Los elementos de memoria se refieren a los hipercubos: $Memoria = [Aa, Ab, Ba, Bb, Cc, Cd, Dc, Dd, Ee, Ef, Fe, Ff]$.

s_1, s_2	*célula*	*Memoria*	a_i	*en la parte*	*ENTONCES*
(P,P)	$(4,4)$	[0 0 0 1 0 0 0 1 0 0 0 1]		[1 1 1 1 0 0 0 1 0 0 0 1]	
	$(4,2)$	[0 0 1 0 0 0 0 1 0 0 0 1]			
	$(2,2)$	[1 0 0 0 0 0 0 1 0 0 0 1]			
	$(2,4)$	[0 1 0 0 0 0 0 1 0 0 0 1]			
(P,N)	$(2,-4)$	[1 0 0 0 0 0 1 0 0 0 1 0]		[1 0 1 1 0 0 1 1 0 0 1 0]	
	$(2,-2)$	[1 0 0 0 0 0 1 0 0 0 1 0]			
	$(4,-4)$	[0 0 0 1 0 0 1 0 0 0 1 0]			
	$(4,-2)$	[0 0 1 0 0 0 0 1 0 0 1 0]			

s_1, s_2	*célula*	*Memoria*	a_i	*en la parte*	*ENTONCES*
(N,P)	$(-4,2)$	[1 0 0 0 0 1 0 0 0 1 0 0]		[1 1 0 0 0 1 0 1 0 1 0 0]	
	$(-4,4)$	[0 1 0 0 0 1 0 0 0 1 0 0]			
	$(-2,2)$	[1 0 0 0 0 0 0 1 0 1 0 0]			
	$(-2,4)$	[0 1 0 0 0 0 0 1 0 1 0 0]			
(N,N)	$(-4,-4)$	[1 0 0 0 1 0 0 0 1 0 0 0]		[1 0 0 0 1 1 1 1 1 0 0 0]	
	$(-4,-2)$	[1 0 0 0 0 1 0 0 1 0 0 0]			
	$(-2,-4)$	[1 0 0 0 0 0 1 0 1 0 0 0]			
	$(-2,-2)$	[1 0 0 0 0 0 0 1 1 0 0 0]			

Como se ha visto en esta sección la red neuronal CMAC presenta una estructura similar a la red con funciones de base radial (FBR), ya que ambos pueden usar la función gaussiana como función de activación en su capa oculta, presentan una rápida convergencia del algoritmo de aprendizaje en relación al MLP. La red FBR es de naturaleza local por lo que computacionalmente consumen menos recursos y menos tiempo, son más fáciles de imple-

mentar por software y hardware, sin embargo existen algunas modificaciones a la red CMAC que la han hecho más poderosa.

Las redes FBR solo están diseñados para abordar problemas estáticos ya que presenta una estructura rígida hacia adelante de tres capas, por otra parte las redes CMAC pueden presentar una recurrencia local, global ó una combinación de ambas lo que permite su uso en la identificación y control de sistemas dinámicos. Al combinar la herramienta de los sistemas difusos y las redes CMAC, se obtiene una red CMAC difusa, este esquema presenta la capacidad de aprendizaje de la red junto con el conocimiento a priori del sistema y una estrategia de control difusa. El problema de la dimensionalidad de la explosión de las reglas se resuelve usando una estructura jerárquica. Estas son algunas de las ventajas de las redes CMAC con respecto a las redes con funciones de base radial.

Capítulo 3

Identificación mediante redes FCMAC

3.1. Preliminares

En 1892 A. M. Lyapunov desarrolló dos métodos en su famosa monografía: *Problema general de la estabilidad del movimiento*, presentó dos métodos para determinar la estabilidad de los sistemas dinámicos descritos mediante ecuaciones diferenciales ordinarias. El primer método se compone de todos los procedimientos en los cuales se usa la forma explícita de la solución de las ecuaciones diferenciales para el análisis. En cambio, el segundo método no requiere de las soluciones de las ecuaciones diferenciales, es decir, mediante el segundo método de Lyapunov se determina la estabilidad de un sistema sin resolver las ecuaciones de estado, lo cuál resulta en un método más sencillo, de manera general el método de Lyapunov es para sistemas autónomos y no autónomos sin perturbaciones del tipo $x\left(k+1\right)=f\left[k,x\left(k\right),0\right]$. Los teoremas de estabilidad de Lyapunov dan condiciones suficientes para la estabilidad y estabilidad asintótica, tambien pueden ser usados para mostrar acotamiento de la solución.

La noción de estabilidad entrada-estado (iss por sus siglas en ingles) es una extensión natural de la estabilidad de Lyapunov para sistemas con entradas. La iss es una de las herramientas más importantes para estudiar la dependencia de la trayectoria de los estados de los sistemas no lineales en tiempo continúo y discreto sobre la magnitud de las entradas, las cuáles pueden representar entradas de control ó perturbaciones con estructura $x\left(k+1\right)=$

$f\left[k, x\left(k\right), u\left(k\right)\right]$, es decir, un sistema es iss si cada trayectoria de los estados que corresponde a un control acotado permenece acotado y la trayectoria eventualmente llega a ser pequeña si la señal de entrada es pequeña independientemente del estado inicial. Para estudiar las propiedades de la estabilidad entrada-estado se considera el siguiente sistema no lineal en tiempo discreto en su representación de espacio de estados:

$$\begin{aligned} x(k+1) &= f\left[x\left(k\right), u\left(k\right)\right] \\ y(k) &= g\left[x\left(k\right)\right] \end{aligned} \tag{3.1}$$

Donde $x\left(k\right) \in \mathbb{R}^n$ es el vector de estados y $u\left(k\right) \in \mathbb{R}^m$ es el vector de entrada. Para algún n, m y para cada instante de tiempo $k \in \mathbb{Z}_+$, f, g son funciones no lineales $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Para cualquier $x \in \mathbb{R}^n$, x^T es la transpuesta y $|x|$ es la norma euclideana. Para una matriz A de $n \times m$, $\|A\|$ es la norma de la matriz. Se emplea $\mathbb{R}, \mathbb{R}_+, \mathbb{Z}, \mathbb{Z}_+$ para denotar al conjunto de los números reales, a los reales no negativos, a los enteros y a los números enteros no negativos respectivamente. Para cada $x_0 \in \mathbb{R}^n$ y cada entrada u, se denota a $x\left(\cdot, x_0, u\right)$ como la trayectoria del sistema (3.1), con el estado inicial $x\left(0\right) = x_0$ y la entrada u. Las entradas ó señales de control $u\left(\cdot\right)$ son funciones de $\mathbb{Z}_+$ a $\mathbb{R}^m$. Se considera el sistema con valores de entrada tomados en un subconjunto restringido $\Omega \subset \mathbb{R}^m$; se utiliza $\mathcal{M}_\otimes$ como el conjunto de valores de control tomados de Ω. Es claro que la trayectoria está definida únicamente sobre $\mathbb{Z}_+$. Se considera que $f\left(0, 0\right) = 0$, es decir, $x_0 = 0$ es un estado de equilibrio del sistema en la entrada 0. Se introducen las definiciones de funciones clase $\mathcal{K}$, $\mathcal{K}_\infty$ y $\mathcal{KL}$ y los conceptos de funciones iss y $Lyapunov - iss$.

Definición 3.1 *Función clase $\mathcal{K}$. Si una función continua $\gamma\left(\cdot\right) : [0, a) \to [0, \infty)$ es estrictamente creciente con $\gamma\left(0\right) = 0$. $\gamma\left(\cdot\right)$ es llamada una función clase $\mathcal{K}$. Si $\gamma\left(s\right)$ es una función clase $\mathcal{K}$, y además no está acotado, i.e. $\lim_{s\to\infty} \gamma\left(s\right) = \infty$ se dice que $\gamma\left(s\right)$ pertenece a la clase $\mathcal{K}_\infty$.*

Definición 3.2 *Función clase $\mathcal{KL}$. Una función continúa $\beta\left(s, t\right) : [0, a) \times [0, \infty) \to [0, \infty)$ es llamada una función clase $\mathcal{KL}$ si el primer argumento de la función $\beta\left(s, \cdot\right)$ es clase $\mathcal{K}$, y el segundo argumento de la función $\beta\left(\cdot, t\right)$ es estrictamente decreciente y $\lim_{t\to\infty} \beta\left(\cdot, t\right) = 0$.*

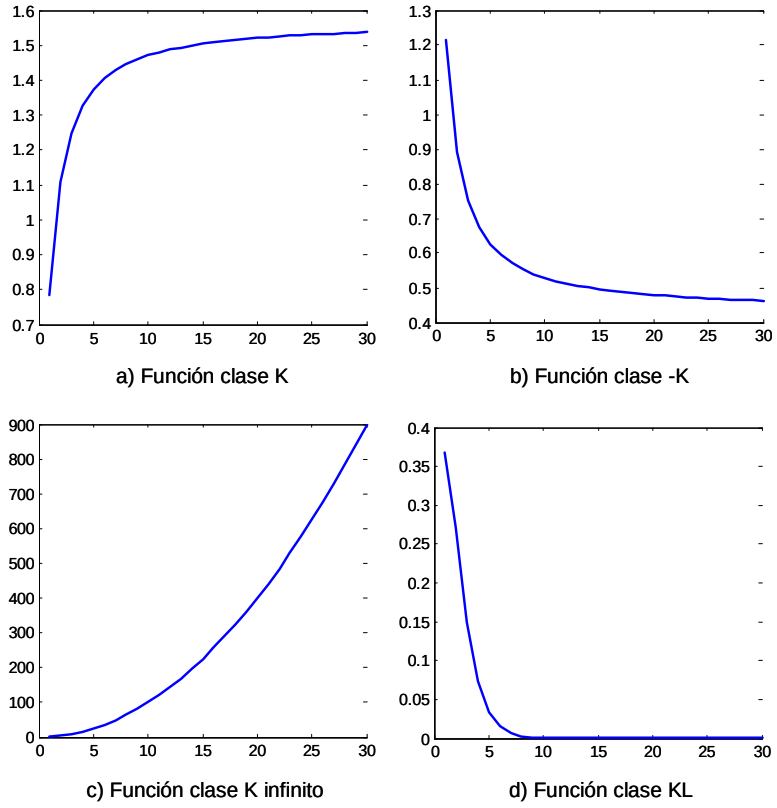

Figura 3.1: Tipos de funciones: a) clase k, b) clase -k, c) clase k-infinito, d) clase kl

La siguientes figuras muestran las funciones: a) clase $\mathcal{K}$, b) clase $\mathcal{K}_\infty$, c) clase $\mathcal{KL}$.

Sean $\gamma_1(\cdot)$ y $\gamma_2(\cdot)$ dos funciones clase $\mathcal{K}$, sobre $[0, a)$. Se expresan $\alpha_1(\cdot), \alpha_2(\cdot)$ como funciones clase $\mathcal{K}_\infty$ y una función β clase $\mathcal{KL}$, la inversa de γ_i se define como γ_i^{-1}, se tienen las siguientes características:

- γ_1^{-1} está definido sobre $[0, \gamma_1(a))$ y pertenece a una clase $\mathcal{K}$.
- α_2^{-1} se define sobre $[0, \infty)$ y pertenece a la clase $\mathcal{K}_\infty$.
- $\gamma_1 \circ \gamma_2$ pertenece a una clase $\mathcal{K}$.

- $\alpha_1 \circ \alpha_2$ pertenece a una clase $\mathcal{K}_\infty$.
- $\sigma(r,s) = \gamma_1(\beta(\gamma_2(r),s))$ pertenecen a la clase $\mathcal{KL}$.

Definición 3.3 *Un sistema representado como (3.1), se dice ser entrada-estado estable si existe una función $\gamma(\cdot)$ de clase $\mathcal{K}$, y una función $\beta(\cdot)$ de clase $\mathcal{KL}$, tal que $u \in L_\infty$, i.e.,* $\sup\|u(k)\| < \infty$. *y cada estado inicial $x_0 \in \mathbb{R}^n$ presenta:*

$$\|x(k, x_0, u(k))\| \leq \beta(\|x_0\|, k) + \gamma(\|u(k)\|) \tag{3.2}$$

Para cada $k \in \mathbb{Z}_+$. Hay que notar que por causalidad, la definición (3.2) puede ser reemplazada por:

$$\|x(k, x_0, u(k))\| \leq \beta(\|x_0\|, k) + \gamma(\|u(k-1)\|) \tag{3.3}$$

Se puede observar en la ecuación (3.3) que la propiedad iss implica que el sistema con entrada 0; $x(k+1) = f[x(k), 0]$ *es globalmente asintóticamente estable (GAS) y que (3.1) es .^{en}trada convergente, estado convergente ", i.e., cada trayectoria del estado $x(k, x_0, u(k))$ va a* 0 *si $u(k)$ va a* 0 *cuando $k \to \infty$, sin embargo lo contrario no es cierto.*

Definición 3.4 *Una función suave $V : \mathbb{R}^n \to \mathbb{R} \geq 0$ es llamada una función de Lyapunov − iss para el sistema (3.1) si las siguientes condiciones permanecen:*

a).- Existen funciones $\alpha_1(\cdot), \alpha_2(\cdot)$ de clase $\mathcal{K}_\infty$ tal que:

$$\alpha_1(x_0) \leq V(x_0) \leq \alpha_2(x_0) \quad \forall x_0 \in \mathbb{R}^n$$

b).- Existe una función $\alpha_3(\cdot)$ de clase $\mathcal{K}_\infty$ y una función $\alpha_4(\cdot)$ de clase $\mathcal{K}$ tal que $\forall x_0 \in \mathbb{R}^n$, cada $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$.

$$V_{k+1} - V_k \leq -\alpha_3(\|x_0(k)\|) + \alpha_4(\|u(k)\|)$$

Una función suave de $Lyapunov - iss$ es una función suave. Es claro que si V es una función de $Lyapunov - iss$ para el sistema (3.1), entonces V es una función de Lyapunov para el sistema

$x(k+1) = f[x(k), 0]$ cuando la entrada $u = 0$, como en la teoria clásica de estabilidad de Lyapunov.

Se considera el sistema $x(k+1) = f[x(k), u(k)]$ donde $u \in \mathcal{M}_{\otimes}$ para algunos subconjuntos compactos $\Omega \subset \mathbb{R}^m$ y f es continua.

Definición 3.5 *Un sistema $x(k+1) = f[x(k), u(k)]$ es globalmente asintóticamente estable si las siguientes dos propiedades permanecen:*

Para cada $\varepsilon > 0$, existe algún $\delta > 0$, tal que $|x(k, x_0, u)| < \varepsilon$ para todo $k \geq 0$, para todo $u \in \mathcal{M}_{\otimes}$, y para todo $|x_0| < \delta$.

La propiedad cuando $\lim_{k\to\infty} |x(k, x_0, u)| = 0$ permanece para todo $x_0 \in \mathbb{R}^n$, $\forall u \in \mathcal{M}_{\otimes}$.

Estas definiciones implican que para el sistema no lineal (3.1), los siguientes términos son equivalentes: a) Entrada-estado estable, b) robusto estable, c) función suave de $Lyapunov - iss$ como se demuestra en [30].

3.1.1. Modelo en el espacio de estados

Un sistema dinámico también puede ser descrito por un modelo en el espacio de estados. El modelo en el espacio de estados de un sistema dinámico no lineal invariante en el tiempo con múltiples entradas múltiples salidas se define a continuación:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k)) \end{aligned} \tag{3.4}$$

donde $x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T$ son los n componentes del vector de estado del sistema, $u(k) = [u_1(k), u_2(k), \ldots, u_r(k)]^T$ es el vector de entradas al sistema y $y(k) = [y_1(k), y_2(k), \ldots, y_m(k)]^T$, es el vector de salida, f, g son los mapeos no lineales estáticos. Si el sistema es lineal, la ecuación (3.4) se convierte en:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{3.5}$$

donde A, B y C son matrices de dimensiones $n \times n$, $n \times r$, y $m \times n$. El esquema de identificación se muestra en la figura 3.2:

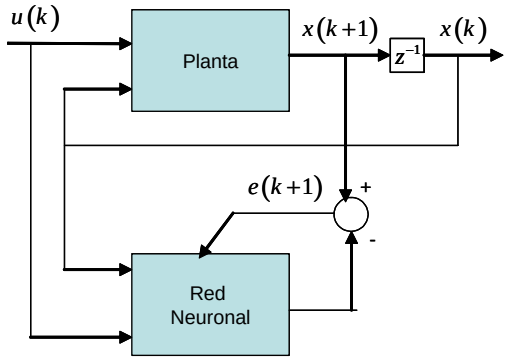

Figura 3.2: Esquema de identificación.

3.2. Identificación mediante redes neuronales CMAC

Las redes neuronales hacia delante han demostrado que presentan resultados aceptables en la identificación y control de sistemas, en donde hay un desconocimiento parcial ó total de la planta y únicamente se dispone de los datos de entrada y salida, esta topología de red en particular realiza un mapeo estático de entrada-salida aproximando funciones continúas no lineales. Sin embargo una desventaja que presentan estas redes es que su aplicación están limitados a problemas estáticos por la misma estructura de la red hacia adelante (feedforward). Para sistemas dinámicos las redes con estructura hacia adelante no son adecuados. Las redes neuronales recurrentes no presentan las desventajas de las redes estáticas y se ha demostrado que son buenos aproximadores de sistemas dinámicos contínuos. Es también conocido, que los sistemas difusos y las redes neuronales están dirigidos a procesar el conocimiento humano, la combinación de ambas técnicas ha presentado una gran diversidad de aplicaciones. En la siguiente sección se diseña una topología de red con recurrencia local y recurrencia global neuro-difusa CMAC analizando las nuevas ventajas que presenta.

3.2.1. Red FCMAC con recurrencia local

La red FCMAC con recurrencia local se presenta en la figura 3.3, está compuesta de *cinco* capas: capa de entrada, capa de fuzzificación, capa de asociación difusa, capa de post-asociación difusa y la capa de salida [8].

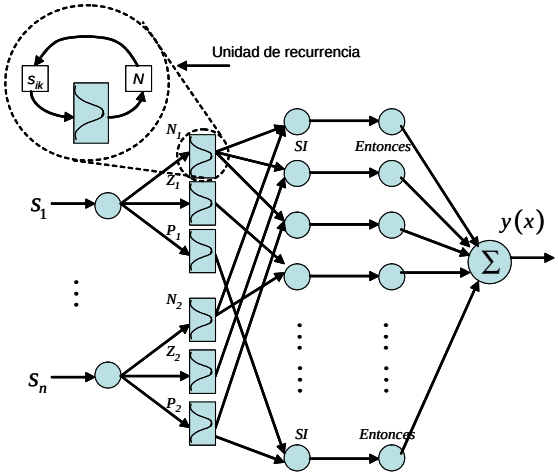

Figura 3.3: Red recurrente local FCMAC

1. La capa de entrada s_i está dada por $s = [s_1, s_2, \ldots, s_n]^T, s \in R^n$.

2. La capa dos presenta una recurrencia local en cada una de las neuronas, donde la función de pertenencia es gaussiana y se representa por:

$$mf_i(k) = s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j} \tag{3.6}$$

donde $\theta_{i,j}$ denota los pesos de cada nodo retroalimentado, $\mu_{A_{i,j}}(k)$ es el término memoria, el cuál almacena información pasada de la red y se encuentra definido como:

$$\mu_{A_j^i}(k) = \exp\left[-\frac{(mf_i(k-1) - c_{i,j})^2}{\sigma_{i,j}^2}\right] \tag{3.7}$$

La expresión 3.6, es la principal diferencia entre la red FCMAC dada por 2.21, y la red con recurrencia local FCMAC (RFCMAC) donde se incluye el elemento de recurrencia. Cada nodo en esta capa presenta tres parámetros a ser ajustados $\theta_{i,j}, c_{i,j}, \sigma_{i,j}$.

3. La capa tres realiza el cálculo de la parte antecedente de la regla difusa. Cada nodo en esta capa realiza la operación de implicación, se utiliza en este trabajo el producto, por lo que se escribe como:

$$\alpha_p = \prod_{i=1}^{n} \lambda_p \left(\mu_{A_j^i} (mf_i) \right), \quad p = 1 \cdots l, \quad j = 1 \cdots m \tag{3.8}$$

Donde p son las veces de asociación, l es el número de asociación y λ es el vector de selección de la memoria de asociación, el cuál se define como:

$$\lambda_p \left(\mu_{A_j^i} (mf_i) \right) = \mu_{A_{j,p}^i} (mf_i) = [0, 0 \cdots 1, 0 \cdots] \begin{bmatrix} \mu_{A_1^i}, \\ \vdots \\ \mu_{A_m^i} \end{bmatrix} \tag{3.9}$$

4. La capa cuatro, tambien conocido como espacio de la memoria de pesos, cálcula la normalización y está dado por:

$$\varphi_p = \frac{\alpha_p}{\sum_{p=1}^{l} \alpha_p} = \frac{\left[\prod_{i=1}^{n} \lambda_p \left(\mu_{A_j^i} (mf_i) \right) \right]}{\left[\sum_{k=1}^{l} \prod_{i=1}^{n} \lambda_p \left(\mu_{A_j^i} (mf_i) \right) \right]} \tag{3.10}$$

5. En la capa de salida se emplea la inferencia difusa Takagi, la salida de la red recurrente FCMAC se expresa como:

$$\hat{y} = \sum_{p=1}^{l} w_p \varphi_p \tag{3.11}$$

Se considera la salida de la red neuronal FCMAC en notación vectorial:

$$\hat{y} = W^T (k) \, \varphi \left[s (k) \right] \tag{3.12}$$

Algoritmo de aprendizaje estable

Se considera una red CMAC, la cuál esta representada por (3.12). El sistema no lineal a ser identificado está representado como (3.1). como se menciono anteriormente, la estabilidad entrada-estado es una extensión de la estabilidad asintótica global para sistemas no lineales, en donde $u(k)$ son entradas de control ó perturbaciones y las salidas son los estados $y(k) = x(k+1)$. i.e.,

$$y(k) = g[x(k)] \tag{3.13}$$

Se diseña un algoritmo de aprendizaje estable tal que la salida $\widehat{y}(k)$ con retroalimentación local de la red FCMAC (3.12) pueda aproximar la salida $y(k)$ de una planta no lineal (3.1).

Se define el error de identificación $e(k)$ como:

$$e(k) = \widehat{y}(k) - y(k) \tag{3.14}$$

y de acuerdo a la teoría de aproximación de funciones de lógica difusa [27], la identificación no lineal de un sistema (3.1) se puede representar como:

$$y(k) = W^*(k)\varphi[mf_i(k)] - \gamma(k) \tag{3.15}$$

donde $W^*(k)$ son los pesos desconocidos del algoritmo de aprendizaje de la red FCMAC que pueden minimizar el error de modelado $\gamma(k)$. De 3.15 y 3.12 se obtiene:

$$\begin{aligned} e(k) &= W^T(k)\varphi[mf_i(k)] - [W^*(k)\varphi[mf_i(k)] - \gamma(k)] \\ e(k) &= \left[W^T(k) - W^*(k)\right]\varphi[mf_i(k)] + \gamma(k) \\ e(k) &= \widetilde{W}(k)\varphi[mf_i(k)] + \gamma(k) \end{aligned} \tag{3.16}$$

donde $\widetilde{W}(k) = W(k) - W^*(k)$. La planta con la estructura (3.1) es $BIBO$ estable (entradas acotadas-salidas acotadas), *i.e.*, $y(k)$ y $u(k)$ en (3.1) son acotados. Las funciones φ, $\gamma(k)$ en (3.15) son acotados. El siguiente teorema desarrollado en el trabajo de tesis presenta el algoritmo del gradiente descendente estable para la identificación no lineal de sistemas mediante redes recurrentes locales FCMAC.

Teorema 3.1 *Se utiliza una red neuronal recurrente FCMAC como el de la figura 3.3 para identificar una planta no lineal (3.1), el algoritmo del gradiente descendente (3.17) presenta una tasa de aprendizaje variante en el tiempo, esto puede hacer el error de identificación* $e(k)$ *acotado.*

$$W(k+1) = W(k) - \eta(k)\, e(k)\, \varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\, \theta_{i,j} \right] \tag{3.17}$$

donde la tasa de aprendizaje $\eta(k)$ *está dado por:*

$$\eta(k) = \frac{\eta}{1 + \left\| \varphi \left[s_i(k) + \mu_{A_j^i}(k-1)\, \theta_{i,j} \right] \right\|^2}; \quad 0 < \eta \leq 1 \tag{3.18}$$

el error de identificación normalizado se define como:

$$e_N(k) = \frac{e(k)}{1 + \max\limits_k \left(\left\| \varphi \left[s_i(k) + \mu_{A_j^i}(k-1)\, \theta_{i,j} \right] \right\|^2 \right)} \tag{3.19}$$

donde (3.19) satisface el siguiente criterio de desempeño:

$$\limsup_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \|e_N(k)\|^2 \leq \overline{\gamma} \tag{3.20}$$

donde $\overline{\gamma}$ *está definido por* $\overline{\gamma} = \max\limits_k \left[\|\gamma(k)\|^2 \right]$.

Demostración. Se selecciona un escalar definido positivo $L(k)$ como:

$$L(k) = \left\| \widetilde{W}(k) \right\|^2$$

Por lo que $L(k+1) = \left\| \widetilde{W}(k+1) \right\|^2$, mediante la ley de actualización (3.17), se tiene:

$$\widetilde{W}(k+1) = \widetilde{W}(k) - \eta(k)\, e(k)\, \varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\, \theta_{i,j} \right]$$

Utilizando las desigualdades $\|a-b\| \geq \|a\| - \|b\|, 2\|ab\| \leq a^2+b^2$, para cualquier a y b. Al utilizar $0 \leq \eta(k) \leq \eta \leq 1$, se obtiene:

$$\begin{aligned}
\Delta L(k) &= L(k+1) - L(k) \\
\Delta L(k) &= \left\|\widetilde{W}(k) - \eta(k)\, e(k)\, \varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2 - \left\|\widetilde{W}(k)\right\|^2 \\
&= \left\|\widetilde{W}(k)\right\|^2 - 2\eta(k) \left\|e(k)\,\widetilde{W}(k)\,\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\| \\
&\quad +\eta^2(k) \left\|e(k)\,\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2 - \left\|\widetilde{W}(k)\right\|^2 \\
&= \eta^2(k) \left\|e(k)\,\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2 \\
&\quad -2\eta\$(k) \left\|e(k)\,\widetilde{W}(k)\,\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\| \\
&= \eta^2(k) \left\|e(k)\,\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2 \\
&\quad -2\eta(k)\|e(k)\| \left\|\widetilde{W}(k)\right\| \left\|\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|
\end{aligned} \tag{3.21}$$

de la ecuación (3.16) se tiene:

$$e(k) - \gamma(k) = \widetilde{W}(k)\,\varphi\left[mf_i(k)\right]$$

y del segundo término de la ecuación (3.21), empleando la desigualdad$\|a-b\| \geq \|a\| - \|b\|$ se tiene:

$$\begin{aligned}
-2\eta(k)\|e(k)\| \left\|\widetilde{W}(k)\right\| \left\|\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,.\theta_{i,j}\right]\right\| &= -2\eta(k)\|e(k)\|\,\|e(k) - \gamma(k)\| \\
-2\eta(k)\left\|e^2(k) - e(k)\,\gamma(k)\right\| &\leq -2\eta(k)\|e(k)\|^2 + 2\eta(k)\|e(k)\|\,\|\gamma(k)\| \\
-2\eta(k)\|e(k)\| \left\|\widetilde{W}(k)\right\| \left\|\varphi^T \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\| &\leq -2\eta(k)\|e(k)\|^2 + 2\eta(k)\|e(k)\|\,\|\gamma(k)\|
\end{aligned}$$

Por la desigualdad $2\|ab\| \leq a^2+b^2$, se tiene $2\eta(k)\|e(k)\,\gamma(k)\| \leq \eta(k)\|e(k)\|^2 + \eta(k)\|\gamma(k)\|^2$, por lo que:

$$\begin{aligned}
&\Delta L(k) = L(k+1) - L(k) \\
&\leq \eta^2(k)\|e(k)\|^2 \left\|\varphi\left(s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2 - \eta(k)\|e(k)\|^2 + \eta(k)\|\gamma(k)\|^2 \\
&\leq -\eta(k)\|e(k)\|^2 + \eta^2(k)\|e(k)\|^2 \left\|\varphi^T\left(s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2 + \eta(k)\|\gamma(k)\|^2 \\
&\leq -\eta(k)\|e(k)\|^2 \left(1 - \eta_k \left\|\varphi^T\left(s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2\right) + \eta(k)\|\gamma(k)\|^2
\end{aligned} \tag{3.22}$$

recordando que $\eta(k) = \dfrac{\eta}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2}$, la ec. (3.22) se puede escribir como:

$$\Delta L(k) \leq -\eta(k)\,\|e(k)\|^2 \left(1 - \left(\frac{\eta}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2}\right)\left(\frac{}{\left\|\varphi^T\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2}\right)\right) + \eta(k)\,\|\gamma(k)\|^2$$

$$\Delta L(k) \leq -\eta(k)\,\|e(k)\|^2 \left(1 - \left(\frac{\eta\left\|\varphi^T\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2}\right)\right) + \eta(k)\,\|\gamma(k)\|^2$$

sustituyendo el valor de $\eta(k)$ en la ecuación anterior:

$$\begin{aligned}\Delta L(k) \;\leq\; & -\|e(k)\|^2 \left(\frac{\eta}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2}\right) \\ & \left(1 - \left(\frac{\eta\left\|\varphi^T\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2}\right)\right) + \eta(k)\,\|\gamma(k)\|^2\end{aligned}$$

$$\Delta L(k) \leq -\|e(k)\|^2 \left(\begin{array}{c} \dfrac{\eta}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2} \\ -\dfrac{\eta}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2}* \\ \left(\dfrac{\eta\left\|\varphi^T\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2}\right) \end{array} \right) + \eta(k)\|\gamma(k)\|^2$$

$$\Delta L(k) \leq -\|e(k)\|^2 \left[\begin{array}{c} \dfrac{\eta}{1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2} \\ -\left(\dfrac{\eta^2\left\|\varphi^T\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2}{\left[1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2\right]^2}\right) \end{array} \right] + \eta(k)\|\gamma(k)\|^2$$

$$\Delta L(k) \leq -\|e(k)\|^2 \left[\left(\frac{\eta-\eta^2\left\|\varphi^T\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2}{\left[1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2\right]^2} \right) \right] + \eta(k)\|\gamma(k)\|^2$$

$$\Delta L(k) \leq -\|e(k)\|^2 \left[\left(\frac{\eta\left(1-\eta\left\|\varphi^T\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2\right)}{\left[1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2\right]^2} \right) \right] + \eta(k)\|\gamma(k)\|^2 \tag{3.23}$$

se define $\pi(k)$ como:

$$\pi(k) = \left(\frac{\eta\left(1-\eta\left\|\varphi\left(s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right)\right\|^2\right)}{\left[1+\left\|\varphi\left[s_i(k)+\mu_{A_j^i}(k-1)\,\theta_{i,j}\right]\right\|^2\right]^2} \right) \tag{3.24}$$

por lo que la ec. (3.23) se escribe en forma simplificada:

$$\Delta L(k) \leq -\pi(k)\|e(k)\|^2 + \eta(k)\|\gamma(k)\|^2 \tag{3.25}$$

se conoce que $0 \leq \left\| \varphi \left(s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j} \right) \right\|^2 \leq \text{máx} \left(\left\| \varphi \left(s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j} \right) \right\|^2 \right)$, y además $\eta(k) \leq \eta$, se tiene:

$$\eta(k) \left(1 - \eta \left\| \varphi \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j} \right] \right\|^2 \right) \geq \frac{\eta}{\left[1 + \text{máx} \left(\left\| \varphi \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j} \right] \right\|^2 \right) \right]^2}$$

la ecuación (3.25) se reescribe como:

$$\Delta L(k) \leq -\pi \left\| e(k) \right\|^2 + \eta \left\| \gamma(k) \right\|^2 \tag{3.26}$$

donde π está definido como:

$$\pi = \frac{\eta}{\left[1 + \text{máx} \left(\left\| \varphi \left[s_i(k) + \mu_{A_j^i}(k-1)\,\theta_{i,j} \right] \right\|^2 \right) \right]^2}$$

Porque $n\,\text{mín}\left(\widetilde{W}_i^2\right) \leq L(k) \leq n\,\text{máx}\left(\widetilde{W}_i^2\right)$, donde $n\,\text{mín}\left(\widetilde{W}_i^2\right)$ y $n\,\text{máx}\left(\widetilde{W}_i^2\right)$ son funciones clase $\mathcal{K}_\infty$, y $\pi \left\| e(k) \right\|^2$ es una función clase $\mathcal{K}_\infty$, $\eta \left\| \gamma(k) \right\|^2$ es una función clase $\mathcal{K}$. $L(k)$ admite una función de Lyapunov ISS. La dinámica del error de identificación es entrada-estado estable. Se conoce que $L(k)$ está en función de $e(k)$ y $\gamma(k)$. Las entradas corresponden al segundo término de (3.26), *i.e.*, el error de modelado $\gamma(k)$. Los estados corresponde al primer término de (3.22), *i.e.*., el error de identificación $e(k)$. Por que la entrada $\gamma(k)$ es acotada y la dinámica es ISS, El estado $e(k)$ es acotado. (3.22) se puede reescribir como:

$$\Delta L(k) \leq -\eta \frac{\left\| e(k) \right\|^2}{\left[1 + \underset{k}{\text{máx}} \left(\left\| \varphi \left[mf(t) \right] \right\|^2 \right) \right]^2} + \eta \overline{\gamma}$$

por lo que:

$$\Delta L(k) \leq -\pi e^2(k) + \eta \gamma_k^2 \leq \pi e^2(k) + \overline{\gamma} \tag{3.27}$$

Resumiendo (3.27) de 1 hasta K, y al utilizar $L(k) > 0$ y L_1 es una constante. Se obtiene:

$$L_K - L_1 \leq -\eta \sum_{k=1}^{K} \left\| e_N(k) \right\|^2 + K \eta \overline{\gamma}$$

por lo que (3.20) es acotado. ■

3.2.2. Red FCMAC con recurrencia global

La arquitectura de la red neuronal con recurrencia global FCMAC se muestra en la figura 3.4, el término z^{-1} indica un retardo en el tiempo, esta topología es un modelo modificado de la red estática FCMAC. A continuación se explica su funcionamiento:

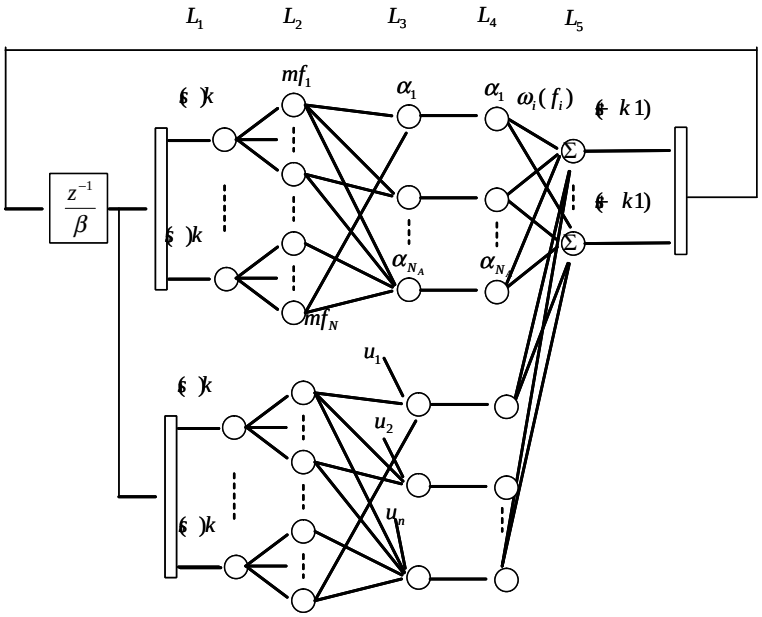

Figura 3.4: Red recurrente global FCMAC.

1. La capa de entrada está dado por $s_r = [s_{r_1}, s_{r_2}, ..., s_{r_n}]^T \in R^n$, donde cada $s_{r_i} = s_i + w_{i,j}\hat{y}(k-1)$.

2. La capa dos presenta la función de pertenencia gaussiana dada por:

$$\mu_{A_j^i}(s_{r_i}) = \exp\left[-\frac{(s_{r_i}(k) - c_{i,j})^2}{\sigma_{i,j}^2}\right]$$

donde los parámetros a ser ajustados en esta capa son $c_{i,j}, \sigma_{i,j}$.

3. La capa tres está dado por:

$$\alpha_p = \prod_{i=1}^{n} \lambda_p \left(\mu_{A_j^i} \left(s_{r_i} \right) \right), \quad p = 1 \cdots l, \quad j = 1 \cdots m \tag{3.28}$$

Donde p son las veces de asociación, l es el número de asociación y λ es el vector de selección de la memoria de asociación, el cuál se define como:

$$\lambda_p \left(\mu_{A_j^i} \left(s_{r_i} \right) \right) = \mu_{A_{j,p}^i} \left(s_{r_i} \right) = [0, 0 \cdots 1, 0 \cdots] \begin{bmatrix} \mu_{A_1^i}, \\ \vdots \\ \mu_{A_m^i} \end{bmatrix} \tag{3.29}$$

4. La capa cuatro cálcula la normalización y está dado por:

$$\varphi_p = \frac{\alpha_p}{\sum_{p=1}^{l} \alpha_p} = \frac{\left[\prod_{i=1}^{n} \lambda_p \left(\mu_{A_j^i} \left(s_{r_i} \right) \right) \right]}{\left[\sum_{p=1}^{l} \prod_{i=1}^{n} \lambda_p \left(\mu_{A_j^i} \left(s_{r_i} \right) \right) \right]} \tag{3.30}$$

5. En la capa de salida se emplea la inferencia difusa, por lo que la parte consecuente de las reglas difusas se definen en función de las variables de entrada y la señal de control. Se escribe la estructura de las reglas:

$$\begin{gathered} R^j : \text{ si } s_1 \text{ es } \mu_{A_j^1} \cdots \text{ y } s_n \text{ es } \mu_{A_j^n} \\ \text{entonces } \beta s_1 (k+1) \text{ es } f_1 (s) \text{ ó } s_1 (k+1) \text{ es } f_2 (s) u \end{gathered} \tag{3.31}$$

Por lo que la salida de la red recurrente global FCMAC se expresa en notación vectorial como:

$$\beta s (k+1) = W_1^T \varphi_1 \left[s_r (k) \right] + W_2^T \varphi_2 \left[s_r (k) \right] u (k) \tag{3.32}$$

Algoritmo de aprendizaje de la red recurrente global FCMAC

Se diseña un algoritmo de aprendizaje estable para que la salida de la red recurrente global FCMAC $\widehat{y}(k)$ siga la salida $y(k)$ de la planta (3.1). El error de identificación se

define como en (3.14). La teoría de aproximación de sistemas difusos y redes neuronales [27], menciona que el proceso no lineal a ser identificado (3.1) se puede representar como:

$$\begin{aligned} \beta s(k+1) &= W_1^{*T}\varphi_1\left[s_r\left(k\right)\right] + W_2^{*T}\varphi_2\left[s_r\left(k\right)\right]u\left(k\right) + \nu\left(k\right) \\ y(k) &= x\left(k+1\right) \end{aligned} \tag{3.33}$$

donde W_1^* y W_2^* son los pesos desconocidos que pueden reducir la dinámica no modelada $\nu\left(k\right)$. El error de identificación $e\left(k\right) = \widehat{y}\left(k\right) - y\left(k\right)$ puede ser representado por (3.32) y (3.33), se obtiene la expresión:

$$e\left(k\right) = \widehat{y}\left(k\right) - y\left(k\right) \tag{3.34}$$

$$\begin{aligned} e\left(k\right) &= W_1^T\varphi_1\left[s_r\left(k\right)\right] + W_2^T\varphi_2\left[s_r\left(k\right)\right]u\left(k\right) \\ &\quad -W_1^{*T}\varphi_1\left[s_r\left(k\right)\right] - W_2^{*T}\varphi_2\left[s_r\left(k\right)\right]u\left(k\right) - \nu\left(k\right) \end{aligned} \tag{3.35}$$

$$\begin{aligned} e\left(k\right) &= \left(W_1^T - W_1^{*T}\right)\varphi_1\left[s_r\left(k\right)\right] + \left(W_2^T - W_2^{*T}\right)\varphi_2\left[s_r\left(k\right)\right]u\left(k\right) - \nu\left(k\right) \\ \beta e\left(k+1\right) &= \tilde{W}_1^T\varphi_1\left[s_r\left(k\right)\right] + \tilde{W}_2^T\varphi_2\left[s_r\left(k\right)\right]u\left(k\right) - \nu\left(k\right) \end{aligned}$$

donde $\widetilde{W_1}\left(k\right) = W_1\left(k\right) - W_1^*$, $\widetilde{W_2}\left(k\right) = W_2\left(k\right) - W_2^*$. Se asume que la planta (3.1) es BIBO estable. φ_k, $\nu\left(k\right)$ en (3.33) es acotado. El siguiente teorema es una aportación del trabajo de tesis. Se demuestra el algoritmo del gradiente descendente estable para la red recurrente global FCMAC.

Teorema 3.2 *Si la red recurrente FCMAC (3.32) es usado para identificar la planta no lineal (3.1), y los eigenvalores de A se seleccionan como* $-1 < \lambda\left(A\right) < 0$*, la siguiente ley de actualización de los pesos hace el error de identificación* $e\left(k\right)$ *acotado (estable en el sentido* L_∞*).*

$$\begin{aligned} W_1\left(k+1\right) &= W_1\left(k\right) - \eta\left(k\right)\varphi_1\left[s_r\left(k\right)\right]e^T\left(k\right) \\ W_2\left(k+1\right) &= W_2\left(k\right) - \eta\left(k\right)\varphi_2\left[s_r\left(k\right)\right]u\left(k\right)e^T\left(k\right) \end{aligned} \tag{3.36}$$

donde la tasa de aprendizaje se define como:

$$\eta\left(k\right) = \begin{cases} \dfrac{\eta}{1 + \left\|\varphi_1\right\|^2 + \left\|\varphi_2 u\right\|^2} & \text{if } \beta\left\|e\left(k+1\right)\right\| \geq \left\|e\left(k\right)\right\| \\ 0 & \text{if } \beta\left\|e\left(k+1\right)\right\| < \left\|e\left(k\right)\right\| \end{cases} ; \quad 0 < \eta \leq 1 \tag{3.37}$$

Demostración. Se selecciona una función de Lyapunov:

$$V(k) = \left\|\widetilde{W}_1(k)\right\|^2 + \left\|\widetilde{W}_2(k)\right\|^2 \tag{3.38}$$

donde $\left\|\widetilde{W}_1(k)\right\|^2 = \sum_{i=1}^{n} \widetilde{w}_1(k)^2 = tr\left\{\widetilde{W}_1^T(k)\,\widetilde{W}_1(k)\right\}$. Del algoritmo de actualización (3.36) se tiene:

$$\begin{aligned} \widetilde{W}_1(k+1) &= \widetilde{W}_1(k) - \eta(k)\,\varphi_1[s_r(k)]\,e^T(k) \\ \widetilde{W}_2(k+1) &= \widetilde{W}_2(k) - \eta(k)\,\varphi_2[s_r(k)]\,e^T(k) \end{aligned} \tag{3.39}$$

por lo que:

$$\begin{aligned} &\Delta V(k) = V(k+1) - V(k) \\ &= \left\|\widetilde{W}_1(k) - \eta(k)\,\varphi_1[s_r(k)]\,e(k)^T\right\|^2 - \left\|\widetilde{W}_1(k)\right\|^2 \\ &+ \left\|\widetilde{W}_2(k) - \eta(k)\,\varphi_2[s_r(k)]\,u(k)\,e^T(k)\right\|^2 - \left\|\widetilde{W}_2(k)\right\|^2 \\ &= \left\|\widetilde{W}_1(k)\right\|^2 - 2\eta(k)\left\|\varphi_1[s_r(k)]\,\widetilde{W}_1(k)\,e(k)^T\right\| + \eta^2(k)\left\|\varphi_1[s_r(k)]\,e(k)^T\right\|^2 - \left\|\widetilde{W}_1(k)\right\|^2 \\ &+ \left\|\widetilde{W}_2(k)\right\|^2 - 2\eta(k)\left\|\varphi_2[s_r(k)]\,u(k)\,\widetilde{W}_2(k)\,e(k)^T\right\| + \eta^2(k)\left\|\varphi_2[s_r(k)]\,e(k)^T\,u(k)\right\|^2 - \left\|\widetilde{W}_2(k)\right\|^2 \\ &= \eta^2(k)\,\|e(k)\|^2\,\|\varphi_1[s_r(k)]\|^2 - 2\eta(k)\left\|\varphi_1\widetilde{W}_1(k)\,e^T(k)\right\| \\ &+\eta^2(k)\,\|e(k)\|^2\,\|\varphi_2[s_r(k)]\,u(k)\|^2 - 2\eta(k)\left\|\varphi_2[s_r(k)]\,u(k)\,\widetilde{W}_2(k)\,e^T(k)\right\| \\ &= \eta^2(k)\,\|e(k)\|^2\,\|\varphi_1[s_r(k)]\|^2 + \eta^2(k)\,\|e(k)\|^2\,\|\varphi_2[s_r(k)]\,u(k)\|^2 \\ &-2\eta(k)\left\|\varphi_1\widetilde{W}_1(k)\,e^T(k)\right\| - 2\eta(k)\left\|\varphi_2[s_r(k)]\,u(k)\,\widetilde{W}_2(k)\,e^T(k)\right\| \end{aligned} \tag{3.40}$$

existe una constante $\beta > 0$, tal que si $\|\beta e(k+1)\| \geq \|e(k)\|$, utilizando (3.34):

$$\beta e(k+1) - Ae(k) + \nu(k) = \tilde{W}_1^T\varphi_1[s_r(k)] + \tilde{W}_2^T\varphi_2[s_r(k)]\,u(k)$$

y $0 \leq \eta(k)$, de los términos negativos de (3.40) se tiene:

$$\begin{aligned}
&-2\eta(k)\left\|\widetilde{W}_1^T(k)\varphi_1 e^T(k)\right\| - 2\eta(k)\left\|\widetilde{W}_2^T(k)\varphi_2[s_r(k)]u(k)e^T(k)\right\| \\
&= -2\eta(k)e^T(k)\left[\left\|\left(\widetilde{W}_1^T(k)\varphi_1\right)\right\| - \left\|\left(\widetilde{W}_2^T(k)\varphi_2[s_r(k)]u(k)\right)\right\|\right] \\
&= -2\eta(k)e^T(k)\left[\|\beta e(k+1) - Ae(k) + \nu(k)\|\right] \\
&\leq -2\eta(k)\left\|e^T(k)\right\|\|\beta e(k+1) - Ae(k) - \nu(k)\| \\
&= -2\eta(k)\left\|e^T(k)\beta e(k+1) - e^T(k)Ae(k) - e^T(k)\nu(k)\right\| \\
&\leq -2\eta(k)\left\|e^T(k)\beta e(k+1)\right\| + 2\eta(k)e^T(k)Ae(k) + 2\eta(k)\left\|e^T(k)\nu(k)\right\|
\end{aligned} \tag{3.41}$$

de la desigualdad $2\|ab\| \leq a^2 + b^2$, se tiene $a = e^T(k), b = \nu(k)$:

$$\begin{aligned}
\leq\ & -2\eta(k)\|e(k)\|^2 + 2\eta(k)\lambda_{\max}(A)\|e(k)\|^2 + \eta(k)\|e(k)\|^2 + \eta(k)\|\nu(k)\|^2 \\
\leq\ & -\eta(k)\|e(k)\|^2 + 2\eta(k)\lambda_{\max}(A)\|e(k)\|^2 + \eta(k)\|\nu(k)\|^2
\end{aligned}$$

de los términos positivos de (3.40) y de $0 < \eta \leq 1$:

$$\begin{aligned}
&\Delta V(k) \leq \eta^2(k)\|e(k)\|^2\|\varphi_1\|^2 + \eta^2(k)\|e(k)\|^2\|\varphi_2 u(k)\|^2 \\
&-\eta(k)\|e(k)\|^2 + 2\eta(k)\lambda_{\max}(A)\|e(k)\|^2 + \eta(k)\|\nu(k)\|^2 \\
&= -\eta(k)\left[(1 - 2\lambda_{\max}(A)) - \eta\frac{\|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2}{1 + \|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2}\right]e^2(k) \\
&+\eta(k)^2\nu^2(k) \leq -\pi e^2(k) + \eta\nu^2(k)
\end{aligned} \tag{3.42}$$

donde $\pi = \dfrac{\eta}{1+\kappa}\left[1 - 2\lambda_{\max}(A) - \dfrac{\kappa}{1+\kappa}\right]$, $\kappa = \underset{k}{\max}\left(\|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2\right)$. Se observa que $-1 < \lambda(A) < 0$, $\pi > 0$

$$n\min\left(\widetilde{W}_i^2\right) \leq V(k) \leq n\max\left(\widetilde{W}_i^2\right)$$

donde $n \times \min\left(\widetilde{W}_i^2\right)$ y $n \times \max\left(\widetilde{W}_i^2\right)$ son funciones de clase $\mathcal{K}_\infty$, y $\pi e^2(k)$ es una función de clase $\mathcal{K}_\infty$, $\eta\nu^2(k)$ es una funcion de clase $\mathcal{K}$. $V(k)$ admite una función suave de Lyapunov-ISS, la dinámica del error de identificación es entrada-estado estable. La "entrada" corresponde al segundo término de la última linea en (3.40), *i.e.*, el error de modelado $\nu(k)$, el "estado" corresponde al primer termino de la última linea en (3.40), *i.e.*, el error

de identificación $e(k)$. Por que la entrada $\nu(k)$ es acotada y la dinámica es ISS, el estado $e(k)$ es acotado.

Si $\beta \|e(k+1)\| < \|e(k)\|$, $\Delta V(k) = 0$. $V(k)$ es constante, $W_1(k)$ es constante. De aqui $\|e(k+1)\| < \frac{1}{\beta}\|e(k)\|$, $\frac{1}{\beta} < 1$, $e(k)$ es acotado. ■

Comentario 3.1 *La condición "$\eta(k) = 0$ if $\beta \|e(k+1)\| < \|e(k)\|$" es la zona muerta. Si β se selecciona muy grande, la zona muerta es pequeña.*

3.2.3. Red FCMAC con recurrencia global-local

Una nueva topología es la red recurrente FCMAC formada por dos retroalimentaciones; una retroalimentación global de la salida con respecto a la entrada y una recurrencia local presente en cada una de las neuronas que conforman la capa *dos* de la red, como se observa en la figura 3.5.

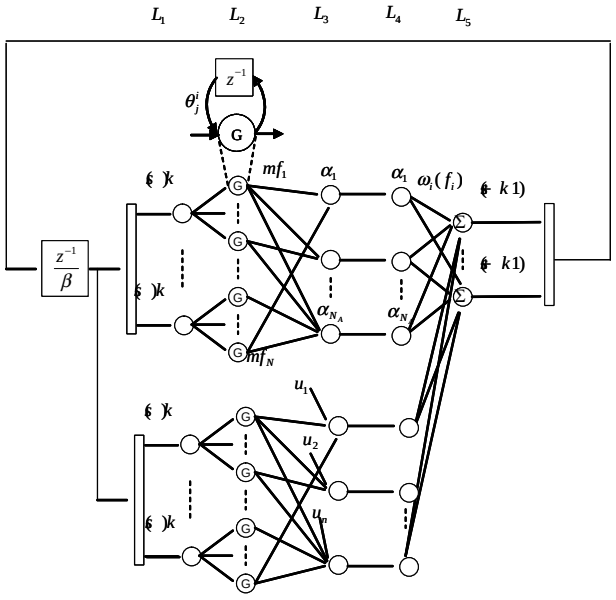

Figura 3.5: Recurrencia Global mas local FCMAC.

La salida en la capa dos de la red recurrente global-local CMAC difusa, está dada por la siguiente expresión:

$$mf_i(k) = \beta s_i(k+1) + \mu_{A_j^i}(k-1)\,\theta_{i,j} \tag{3.43}$$

donde $\theta_{i,j}$ son los pesos de la neurona con recurrencia y $\beta s_i(k+1)$ es la salida de la red recurrente FCMAC. La salida de la red puede ser expresada como:

$$\begin{aligned} \beta s(k+1) &= \sum_{i=1}^{l} w_{1,i}\varphi_{1,i}[mf_i(k)] + \sum_{i=1}^{l} w_{2,i}\varphi_{2,i}[mf_i(k)]\,u(k) \\ \text{ó}\quad \beta s(k+1) &= W_1^T\varphi_1[mf(k)] + W_2^T\varphi_2[mf(k)]\,u(k) \end{aligned} \tag{3.44}$$

W_j $(j = 1, 2)$ denota los pesos entrenables de la red, $\varphi_j(x)$ es la función definida como:

$$\varphi_p = \frac{\prod_{i=1}^{n}\lambda_p\left(mf_j^i\right)}{\sum_{p=1}^{l}\prod_{i=1}^{n}\lambda_p\left(mf_j^i\right)} \tag{3.45}$$

donde $l(p = 1, ..., l)$ es el número de asociación. La función φ_p de la retroalimentación global FCMAC (3.43) es conocida, solo los pesos necesitan ser actualizados para la identificación de sistemas. Se diseña un algoritmo de aprendizaje estable, tal que la salida $s(k)$ de la red neuronal recurrente CMAC difusa (3.44) pueda aproximar la salida $y(k)$ de una planta no lineal (3.1). Se define el vector del error de identificación $e(k)$ como $e(k) = s(k) - y(k)$. De acuerdo a la teoría de aproximación de funciones de lógica difusa y redes neuronales [27], el proceso no lineal a ser identificado (3.1) se puede representar como:

$$\beta s(k+1) = As(k) + W_1^{*T}\varphi_1[s(k)] + W_2^{*T}\varphi_2[s(k)]\,u(k) + \nu(k) \tag{3.46}$$

donde W_1^* y W_2^* son los pesos desconocidos, los cuáles pueden minimizar la dinámica no modelada $\nu(k)$. El error de identificación se puede representar por (3.73) y (3.46),

$$\beta e_p(k+1) = Ae_p(k) + \widetilde{W}_1(k)\,\varphi_1[s(k)] + \widetilde{W}_2^T\varphi_2[s(k)]\,u(k) - \nu(k) \tag{3.47}$$

donde $\widetilde{W}_1(k) = W_1(k) - W_1^*$, $\widetilde{W}_2(k) = W_2(k) - W_2^*$. En este trabajo solo se aborda el problema de la identificación en lazo abierto, se asume que la planta (3.1) presenta entradas acotadas, salidas acotadas, es (BIBO) estable, *i.e.*, $y(k)$ y $u(k)$ en (3.1) son acotados. Por las cotas de las funciones φ_p, $\nu(k)$ en (3.46) es acotado. El siguiente teorema desarrollado en el trabajo de tesis muestra el algoritmo del gradiente descendente estable.

Teorema 3.3 *Si la red neuronal con recurrencia global CMAC difusa (3.44) es utilizada para identificar una planta no lineal (3.1) y los eigenvalores de A se selecionan como* $-1 < \lambda(A) < 0$. *La siguiente ley de adaptación del gradiente, puede hacer el error de identificación* $e(k)$ *acotado (estable en un sentido* L_∞*).*

$$\begin{aligned} W_1(k+1) &= W_1(k) - \eta(k)\,\varphi_1[x(k)]\,e^T(k) \\ W_2(k+1) &= W_2(k) - \eta(k)\,\varphi_2[x(k)]\,u(k)\,e^T(k) \end{aligned} \tag{3.48}$$

donde $\eta(k)$ *satisface*

$$\eta(k) = \begin{cases} \dfrac{\eta}{1 + \|\varphi_1\|^2 + \|\varphi_2 u\|^2} & si\ \beta\|e(k+1)\| \geq \|e(k)\| \\ 0 & si\ \beta\|e(k+1)\| < \|e(k)\| \end{cases}$$

$0 < \eta \leq 1$.

Demostración. Se seleciona una función de Lyapunov como:

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

donde $\left\|\widetilde{W}_1(k)\right\|^2 = \sum_{i=1}^{n} \widetilde{w}_1(k)^2 = tr\left\{\widetilde{W}_1^T(k)\,\widetilde{W}_1(k)\right\}$. De la ley de actualización (3.48)

$$\widetilde{W}_1(k+1) = \widetilde{W}_1(k) - \eta(k)\,\varphi_1[x(k)]\,e^T(k)$$

Por lo que:

$$\begin{aligned} &\Delta V(k) = V(k+1) - V(k) \\ &= \left\|\widetilde{W}_1(k) - \eta(k)\,\varphi_1 e(k)^T\right\|^2 - \left\|\widetilde{W}_1(k)\right\|^2 \\ &+ \left\|\widetilde{W}_2(k) - \eta(k)\,\varphi_2 u(k)\,e^T(k)\right\|^2 - \left\|\widetilde{W}_2(k)\right\|^2 \\ &= \eta^2(k)\,\|e(k)\|^2\,\|\varphi_1\|^2 - 2\eta(k)\left\|\varphi_1\widetilde{W}_1(k)\,e^T(k)\right\| \\ &+ \eta^2(k)\,\|e(k)\|^2\,\|\varphi_2 u(k)\|^2 - 2\eta(k)\left\|\varphi_2 u(k)\,\widetilde{W}_2(k)\,e^T(k)\right\| \end{aligned} \tag{3.49}$$

Existe una constante $\beta > 0$, tal que:

Si $\|\beta e(k+1)\| \geq \|e(k)\|$, utilizando (3.47) y $\eta(k) \geq 0$,

$$\begin{aligned}
&-2\eta(k)\left\|\varphi_1\widetilde{W}_1(k)\,e^T(k)\right\| - 2\eta(k)\left\|\varphi_2 u(k)\,\widetilde{W}_2(k)\,e^T(k)\right\| \\
&\leq -2\eta(k)\left\|e^T(k)\right\|\|\beta e(k+1) - Ae(k) - \nu(k)\| \\
&= -2\eta(k)\left\|\begin{array}{c} e^T(k)\,\beta e(k+1) \\ -e^T(k)\,Ae(k) - e^T(k)\,\nu(k) \end{array}\right\| \\
&\leq -2\eta(k)\left\|e^T(k)\,\beta e(k+1)\right\| + 2\eta(k)\,e^T(k)\,Ae(k) + 2\eta(k)\left\|e^T(k)\,\nu(k)\right\| \\
&\leq -2\eta(k)\|e(k)\|^2 + 2\eta(k)\,\lambda_{\max}(A)\|e(k)\|^2 + \eta(k)\|e(k)\|^2 + \eta(k)\|\nu(k)\|^2
\end{aligned} \tag{3.50}$$

De aquí $0 < \eta \leq 1$

$$\begin{aligned}
&\Delta V(k) \leq \eta^2(k)\|e(k)\|^2\|\varphi_1\|^2 + \eta^2(k)\|e(k)\|^2\|\varphi_2 u(k)\|^2 \\
&-\eta(k)\|e(k)\|^2 + 2\eta(k)\,\lambda_{\max}(A)\|e(k)\|^2 + \eta(k)\|\nu(k)\|^2 \\
&= -\eta(k)\left[\begin{array}{c} (1 - 2\lambda_{\max}(A)) \\ -\eta\dfrac{\|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2}{1 + \|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2} \end{array}\right] e^2(k) + \eta^2(k)\,\nu^2(k) \leq -\pi e^2(k) + \eta\nu^2(k)
\end{aligned} \tag{3.51}$$

donde $\pi = \dfrac{\eta}{1+\kappa}\left[1 - 2\lambda_{\max}(A) - \dfrac{\kappa}{1+\kappa}\right]$, $\kappa = \max\limits_k\left(\|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2\right)$. De aquí $-1 < \lambda(A) < 0$, $\pi > 0$

$$n\,\text{mín}\left(\widetilde{w}_i^2\right) \leq V(k) \leq n\,\text{máx}\left(\widetilde{w}_i^2\right)$$

donde $n \times \text{mín}\left(\widetilde{w}_i^2\right)$ y $n \times \text{máx}\left(\widetilde{w}_i^2\right)$ son funciones de clase $\mathcal{K}_\infty$, y $\pi e^2(k)$ es una función clase $\mathcal{K}_\infty$, $\eta\nu^2(k)$ es una función clase $\mathcal{K}$, tal que $V(k)$ admite una función suave de Lyapunov, La dinámica del error de identificación es entrada-estado estable. La entrada corresponde al segundo término de la última linea en (3.51), *i.e.*, el error de modelado $\nu(k)$, Los estados corresponden al primer término en la última linea de (3.51), *i.e.*, el error de identificación $e(k)$. Por que la entrada $\nu(k)$ es acotado y la dinámica es ISS, el estado $e(k)$ es acotado.

Si $\beta\|e(k+1)\| < \|e(k)\|$, $\Delta V(k) = 0$. $V(k)$ es constante, $W_1(k)$ es constante. De aquí $\|e(k+1)\| < \frac{1}{\beta}\|e(k)\|$, $\frac{1}{\beta} < 1$, $e(k)$ es acotado. ■

Comentario 3.2 *La condición "$\eta(k) = 0$ si $\beta\|e(k+1)\| < \|e(k)\|$" es la zona muerta. Si β es seleccionada con un valor muy grande, la zona muerta será muy pequeña.*

3.3. Redes jerárquicas FCMAC

3.3.1. Representación de una función con estructura jerárquica

Se introduce primero el concepto de la estructura jerárquica natural para funciones continuas. Un sistema con una estructura jerárquica de dos niveles se muestra en la figura 3.6, para este sistema la representación matemática general está dado por: $y = G(s_1, s_2, s_3, s_4, s_5)$. Sin embargo la estructura jerárquica de la figura 3.6, puede también ser representado por:

$$y = g_1(y_1, y_2, s_5), y_1 = g_{21}(s_1, s_2), y_2 = g_{22}(s_3, s_4) \tag{3.52}$$

de otra manera

$$G(s_1, s_2, s_3, s_4, s_5) = g_1[g_{21}(s_1, s_2), g_{22}(s_3, s_4), s_5] \tag{3.53}$$

Entonces para una función dada $y = G(s_1, s_2, s_3, s_4, s_5)$, existen funciones g_1, g_{21}, g_{22} tales que $G(s_1, s_2, s_3, s_4, s_5) = g_1[g_{21}(s_1, s_2), g_{22}(s_3, s_4), s_5]$, esta función puede ser representada por una estructura jerárquica como se muestra en la figura (3.6), en esta representación y_1, y_2 ya no poseen un significado físico, en algunos casos pueden ser considerados como variables de estado.

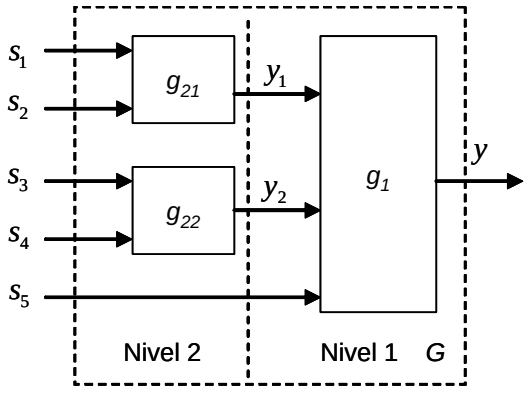

Figura 3.6: Estructura jerárquica con 2 niveles.

Para el caso mencionado anteriormente cualquier estructura física permite a un sistema dado ser representado como una estructura jerárquica ó la estructura matemática de una función puede ser descompuesta como una estructura jerárquica, se puede decir que la función presenta una estructura jerárquica natural [29].

3.3.2. Construcción de la red jerárquica FCMAC

En los esquemas de aproximación neuro-difuso estándar, el número de reglas difusas crece exponencialmente con el número de variables de entrada, específicamente un sistema de lógica difusa con una sola salida y, con n variables de entrada y m funciones de pertenencia, definidas para cada variable de entrada requiere m^n número de reglas difusas [12], esto es, si se tiene 4 variables de entrada con sus respectivas funciones de pertenencia, en este caso 5, entonces necesita $5^4 = 625$ número de reglas difusas, el número de reglas crece exponencialmente. En general los sistemas neuro-difusos estándar presentan el problema de la dimensionalidad, el cuál puede ser vista desde diferentes perspectivas:

- La primera se refiere a la dimensionalidad de las reglas: El número total de reglas difusas se incrementa exponencialmente con el número de variables de entrada.
- La segunda es la dimensionalidad de los parámetros: El número total de parámetros en las fórmulas matemáticas de los sistemas difusos se incrementa exponencialmente con el número de variables de entrada.
- La tercera es la dimensionalidad de los datos ó información: El número de datos ó conjunto de conocimientos requeridos para la identificación de sistemas difusos se incrementa exponencialmente con el número de las variables de entrada [23].

Para reducir el número de reglas involucradas surge la idea de sistemas con estructura jerárquica HFS (Hierarchical Fuzzy Systems), el cuál fue desarrollado principalmente para sistema difusos como se muestra en la figura 3.8. Se pueden observar las diferencias entre las figuras 3.7 y 3.8. La HFS reduce significativamente el número total de reglas difusas involucradas. En los sistemas difusos jerárquicos, las salidas de cada unidad FLU (Fuzzy Logic

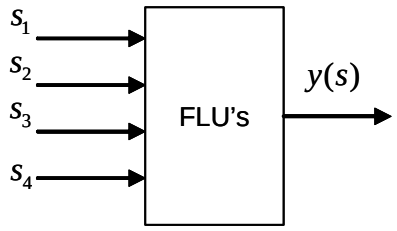

Figura 3.7: Sistema difuso convencional con una sola capa.

Unit) en la capa previa son usados como variables lingüisticas de entrada a la siguiente capa. Las salidas intermedias, sin embargo son de naturaleza artificial y en muchos casos no poseen significado físico. Si se utilizan como las variables de la entrada de la capa siguiente, entonces las reglas difusas involucradas en la mitad de la estructura jerárquica tienen poco significado físico y consecuentemente son difíciles de diseñar. Este fenómeno llega a ser importante cuando el número de capas crece significativamente [27].

Se considera un sistema jerárquico FCMAC (HFCMAC), como se muestra en la figura 3.8, donde la salida de cada bloque $FCMAC_i$ en el nivel *dos*, está dado por:

$$\hat{y}_i = W_i^T \varphi_i^j \left(s_i^j\right) \tag{3.54}$$

donde el subíndice i indica el número de salida y de las redes FCMAC de dimension dos ubicados en el segundo nivel. El superíndice j indica el número de nivel en el que se encuentra cada bloque, todos los bloques de redes FCMAC presentan *dos* entradas. La salida de la red FCMAC en el nivel *uno* está dado por:

$$\hat{y} = W_i^T \varphi_i^j \left(y_i^j\right) \tag{3.55}$$

Considere el siguiente sistema no lineal en tiempo discreto a ser identificado

$$\begin{aligned} &y(k) = h\left[x\left(k\right)\right] \\ &= h\left[y\left(k-1\right), y\left(k-2\right), \cdots u\left(k-1\right), u\left(k-2\right), \cdots\right] \end{aligned} \tag{3.56}$$

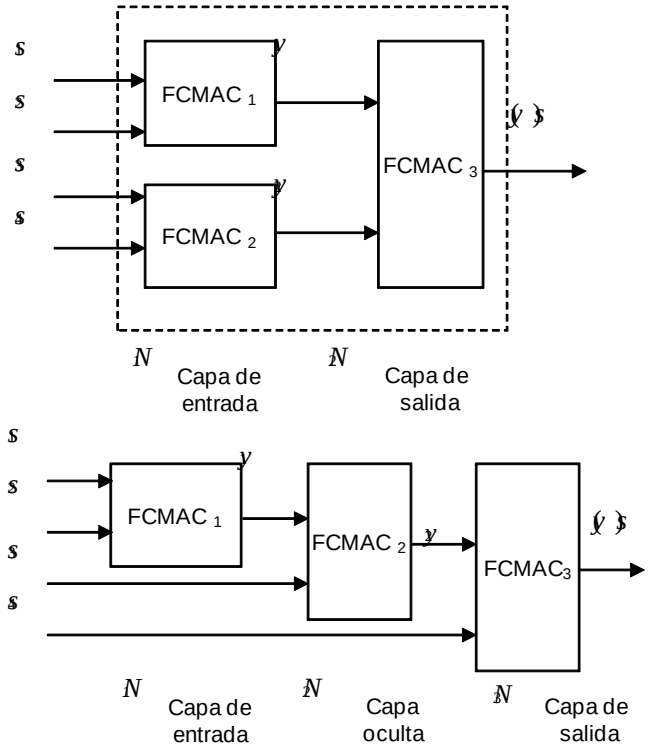

Figura 3.8: Sistema difuso con estructura jerárquica.

donde

$$x(k) = [y(k-1), y(k-2), \cdots u(k), u(k-1), \cdots]^T \tag{3.57}$$

Una red FCMAC puede ser representado como (3.55), el cuál fue explicado a detalle en la sección anterior. Para la red FCMAC el espacio de memoria se incrementa exponencialmente al aumentar el número de variables de entrada, este es un problema serio al utilizar redes FCMAC convencionales, el tamaño de la memoria se dispara exponencialmente. Un método alternativo para minimizar esta dificultad es utilizar redes HFCMAC, la cuál tiene la característica que el espacio de memoria se incrementa linealmente al aumentar el número de variables de entrada.

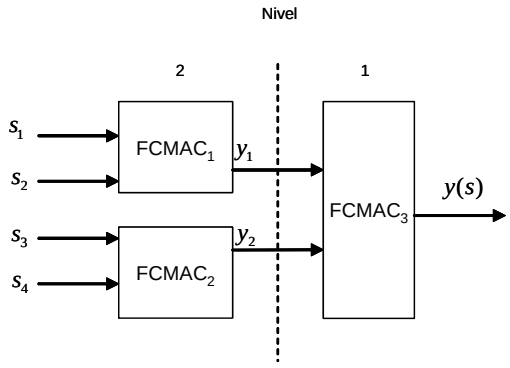

Figura 3.9: Sistema Jerárquico FCMAC.

Se utiliza el siguiente ejemplo para explicar como se aplica la técnica de retropropagación del error para redes HFCMAC. La salida de cada bloque en la red HFCMAC está dado por:

$$\hat{y}_r = \sum_{h=1}^{l} W_{r,p}\varphi_{r,h}, \quad r = 1, 2, 3 \tag{3.58}$$

Las salidas de cada bloque de la red HFCMAC en la capa previa es usada como variables de entrada (variables intermedias) a la siguiente capa. Las salidas intermedias son de naturaleza artificial y en la mayoría de los casos no posee significado físico, lo cuál involucra reglas difusas intermedias en la estructura jerárquica con poco significado físico y consecuentemente difíciles de diseñar, este fenómeno llega a ser importante cuando el número de niveles jerárquicos crece, esto es una de las principales desventajas de esta representación.

A continuación se explica el entrenamiento de cada bloque de la red, se define el índice de desempeño como:

$$J = \frac{1}{2}e_3^2, \quad e_3 = \hat{y}_3 - y \tag{3.59}$$

De la figura 3.9, el algoritmo de aprendizaje para el bloque $FCMAC_3$ es:

$$W_{3,h}\left(k+1\right) = W_{3,h}\left(k\right) - \eta\varphi_{3,h}\left(k\right)e_3\left(k\right) \tag{3.60}$$

Donde $\eta > 0$ es la tasa de aprendizaje, para el subsistema $FCMAC_2$ se requiere actualizar $w_{2,h}(k)$ se puede calcular:

$$\frac{\partial J}{\partial W_{2,h}} = \frac{\partial J}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial \hat{y}_2}\frac{\partial \hat{y}_2}{\partial W_{2,h}} \tag{3.61}$$

De la figura 3.9, se conoce $\frac{\partial \hat{y}_3}{\partial \hat{y}_2}$ que corresponde a $s_{3,2}(k)$, tal que:

$$\begin{aligned}
\frac{\partial J}{\partial \hat{y}_3} &= \hat{y}_3 - y = e_3(k) \\
\frac{\partial \hat{y}_3}{\partial \hat{y}_2} &= \frac{\partial \hat{y}_3}{\partial z_{3,h}}\frac{\partial z_{3,h}}{\partial \hat{y}_2} = \left[\frac{a_3}{b_3^2} - \frac{W_{3,h}}{b_3}\right] z_{3,h}\left[2\frac{\hat{y}_2 - c_{3,2}^i}{\left(\sigma_{3,2}^i\right)}\right] \\
\frac{\partial \hat{y}_2}{\partial W_{2,h}} &= \varphi_{2,h} = \frac{z_{3,h}}{b_2}
\end{aligned}$$

Donde $z_{3,h} = \prod_{i=1}^{n}\lambda_h\left(\mu_{A_j^i}\right)$, $a_3 = \sum_{h=1}^{l} w_{3,h}z_{3,h}$, $b_3 = \sum_{h=1}^{l} z_{3,h}$. el algoritmo de aprendizaje del gradiente para $w_{2,p}$ es:

$$\begin{aligned}
W_{2,h}(k+1) &= W_{2,h}(k) - \eta\varphi_{2,h}(k)\frac{z_{2,h}}{b_2}2\frac{\hat{y}_3 - w_{3,h}}{(b_3)}z_{3,h}\frac{\hat{y}_2 - c_{3,2}^i}{\left(\sigma_{3,2}^i\right)}e_3(k) \\
&= W_{2,h}(k) - \eta\varphi_{2,h}(k)\,e_2(k)
\end{aligned}$$

donde:

$$e_2(k) = 2\frac{\hat{y}_3 - w_{3,h}}{(b_3)}z_{3,h}\frac{\hat{y}_2 - c_{3,2}^i}{\left(\sigma_{3,2}^i\right)^2}e_3(k) = \frac{\partial \hat{y}_3}{\partial \hat{y}_2}e_3(k)$$

de manera similar se obtiene para

$$\begin{aligned}
e_1(k) &= \frac{\partial \hat{y}_3}{\partial \hat{y}_1}e_3(k), \quad \frac{\partial \hat{y}_3}{\partial \hat{y}_1} = 2\frac{\hat{y}_3 - w_3^i}{b_3}z_{3,h}\frac{y_2 - c_{3,1}^i}{\left(\sigma_{3,1}^i\right)^2} \\
W_{1,h}(k+1) &= W_{1,h}(k) - \eta\varphi_{1,h}(k)\,e_1(k)
\end{aligned}$$

Para el caso general el entrenamiento es como sigue:

De acuerdo a la estructura de la red neuronal HFCMAC, se calcula la salida de cada sub-bloque de la red dado por (3.54). Se calcula el error para cada bloque, se inicia con el último bloque, el error de identificación es:

$$e_o(k) = \hat{y}_o(k) - y(k) \tag{3.62}$$

Donde $e_o(k)$ es el error de identificación, $\hat{y}_o(k)$ es la salida del último bloque de la red jerárquica FCMAC, $y(k)$ es la salida de la planta, se retropropaga el error y se calcula el error para el bloque p, (definido como e_p) para formar el bloque q (definido como e_q). Se emplea la regla de la cadena:

$$e_p(k) = 2\frac{\hat{y}_q - w_q^i}{b_q} z_{q,h} \frac{\hat{y}_p - c_{q,p}^i}{\left(\sigma_{q,p}^i\right)^2} e_q(k) \tag{3.63}$$

El entrenamiento de la función gaussiana (funciones de membresia en la premisa y en la parte consecuente), para cada bloque independientemente, para el $p-ésimo$ bloque el algoritmo de retropropagación es:

$$W_{p,h}(k+1) = W_{p,h}(k) - \eta\varphi_{p,h}(k)\, e_p(k) \tag{3.64}$$

Algoritmo de aprendizaje estable

Se asume que la función $\varphi_{r,h}$ de la red HFCMAC para cada bloque es conocida, solo se necesitan actualizar los pesos para identificar el sistema, se diseña el algoritmo de aprendizaje estable tal que la salida $\hat{y}(k)$ de la red CMAC difusa, pueda seguir la salida de la planta no lineal $y(k)$, se define el vector de error de identificación $e(k)$ como:

$$e(k) = \hat{y}(k) - y(k) \tag{3.65}$$

De la ec. 3.63 la señal $e(k)$ puede ser propagada para cada sub-bloque, llamado $e_p(k)$, existe una salida virtual de la planta $y_p(k)$ la cuál corresponde a la salida del sub bloque $\hat{y}_p(k)$, así que:

$$e_p(k) = \hat{y}_p(k) - y_p(k)$$

para el $p-ésimo$ bloque, se asume que la planta no lineal puede ser expresado por alguna función. De acuerdo a las teorías de aproximación de lógica difusa y redes neuronales, el proceso no lineal a identificar puede ser representado como:

$$y_p(k) = W^*(k)\,\varphi[s(k)] - \gamma(k) \tag{3.66}$$

Donde $W^*(k)$ son los pesos desconocidos los cuáles minimizan las dinámicas no modeladas $\gamma(t)$. El error de identificación puede ser representado por (3.65) y (3.66):

$$e_p(k) = \tilde{W}(k)\,\varphi[s(k)] + \gamma(k) \tag{3.67}$$

Donde $\tilde{W}(k) = W(k) - W^*(k)$. Para la identificación de la planta se realizará en lazo abierto, se asume que la planta (3.56) presenta las entradas y salidas acotadas estables (BIBO), *i.e.*, $y(k)$ y $u(k)$ en (3.56) son acotados. Las funciónes $\varphi, \gamma(k)$ en (3.66) son acotadas, el siguiente teorema desarrollado en el trabajo de tesis establece el algoritmo de aprendizaje del gradiente descendente estable para el modelado con sistemas neuro-difusos.

Teorema 3.4 *Si se utiliza una red neuronal jerárquica CMAC difusa para identificar una planta no lineal (3.56), el algoritmo del gradiente descendente (3.64) con una tasa de aprendizaje con tiempo variable se presenta a continuación y puede hacer el error de identificación* $e(k)$ *acotado.*

$$W(k+1) = W(k) - \eta(k)\,e_p(k)\,\varphi^T[s(k)] \tag{3.68}$$

donde el escalar $\eta(k) = \frac{\eta}{1+\|\varphi[s(k)]\|^2}, \quad 0 \leq \eta \leq 1$. El error de identificación normalizado está dado por

$$e_N(k) = \frac{e(k)}{1 + \max\limits_k \|\varphi[s(k)]\|^2}$$

satisface el siguiente comportamiento promedio

$$\lim_{T\to\infty}\sup\frac{1}{T}\sum_{k=1}^{T}\left\|e_N\left(k\right)\right\|^2\leq\bar{\gamma} \tag{3.69}$$

donde $\bar{\gamma}=\max_k\left\|\gamma\left(k\right)\right\|^2$

Demostración. Se selecciona un escalar definido positivo $L\left(k\right)$ como:

$$L\left(k\right)=\left\|\tilde{W}\left(k\right)\right\|^2$$

por la actualización de los pesos (3.68) se tiene:

$$\tilde{W}\left(k+1\right)=\tilde{W}\left(k\right)-\eta\left(k\right)e\left(k\right)\varphi^T\left[s\left(k\right)\right]$$

utilizando la desigualdad:

$$\left\|a-b\right\|\geq\left\|a\right\|-\left\|b\right\|,2\left\|ab\right\|\leq a^2+b^2$$

para cualquier a,b. Al utilizar (3.67) y $0\leq\eta\left(k\right)\leq\eta\leq1$, se tiene:

$$\begin{aligned}
\Delta L\left(k\right) &= L\left(k+1\right)-L\left(k\right)=\left\|\tilde{W}\left(k\right)-\eta\left(k\right)e\left(k\right)\varphi^T\left[s\right]\right\|^2-\left\|\tilde{W}\left(k\right)\right\|^2 \\
&= \left\|\tilde{W}\left(k\right)\right\|^2-2\eta\left(k\right)\left\|e\left(k\right)\varphi^T\left(s\right)\tilde{W}\left(k\right)\right\|+\eta^2\left(k\right)\left\|e\left(k\right)\varphi\left[s\left(k\right)\right]\right\|^2-\left\|\tilde{W}\left(k\right)\right\|^2 \\
&= \eta^2\left(k\right)\left\|e\left(k\right)\right\|^2\left\|\varphi\left[s\left(k\right)\right]\right\|^2-2\eta\left(k\right)\left\|e\left(k\right)\left[e\left(k\right)-\gamma\left(k\right)\right]\right\| \\
&\leq \eta^2\left(k\right)\left\|e\left(k\right)\right\|^2\left\|\varphi\left[s\left(k\right)\right]\right\|^2-2\eta\left(k\right)\left\|e\left(k\right)\right\|^2+2\eta\left(k\right)\left\|e\left(k\right)\gamma\left(k\right)\right\| \\
&\leq \eta^2\left(k\right)\left\|e\left(k\right)\right\|^2\left\|\varphi\left[s\left(k\right)\right]\right\|^2-2\eta\left(k\right)\left\|e\left(k\right)\right\|^2+\eta\left(k\right)\left\|e\left(k\right)\right\|^2+\eta\left(k\right)\left\|\gamma\left(k\right)\right\|^2 \\
&= -\eta\left(k\right)\left\|e\left(k\right)\right\|^2\left(1-\eta_k\left\|\varphi^T\left[s\right]\right\|^2\right)+\eta\left(k\right)\left\|\gamma\left(k\right)\right\|^2
\end{aligned} \tag{3.70}$$

de aqui $\eta\left(k\right)=\frac{\eta}{1+\left\|\varphi[s(k)]\right\|^2}$, $\eta\left(k\right)\left(1-\eta\left(k\right)\left\|\varphi\left[s\left(k\right)\right]\right\|^2\right)=\eta\left(k\right)\left(1-\frac{\eta}{1+\left\|\varphi[s(k)]\right\|^2}\left\|\varphi\left[s\left(k\right)\right]\right\|^2\right)$
$\geq\eta\left(k\right)\left(1-\eta\frac{\max_k\left(\left\|\varphi[s(k)]\right\|^2\right)}{1+\max_k\left(\left\|\varphi[s(k)]\right\|^2\right)}\right)\geq\eta\left(k\right)\left(1-\frac{\max_k\left(\left\|\varphi[s(k)]\right\|^2\right)}{1+\max_k\left(\left\|\varphi[s(k)]\right\|^2\right)}\right)$
$=\frac{\eta(k)}{1+\max_k\left(\left\|\varphi[s(k)]\right\|^2\right)}\geq\frac{\eta}{\left[1+\max_k\left(\left\|\varphi[s(k)]\right\|^2\right)\right]^2}$

así

$$\Delta L\left(k\right)\leq-\pi\left\|e\left(k\right)\right\|^2+\eta\left\|\gamma\left(k\right)\right\|^2 \tag{3.71}$$

donde π está definido como $\pi = \frac{\eta}{\left[1+\max_k\left(\|\varphi[s(k)]\|^2\right)\right]^2}$, porque $n\,\text{mín}\,(\tilde{w}_i^2) \leq L(k) \leq n\,\text{máx}\,(\tilde{w}_i^2)$

Donde $n\,\text{mín}\,(\tilde{w}_i^2)$ y $n\,\text{máx}\,(\tilde{w}_i^2)$ son funciones κ_∞ y $\pi\,\|e(k)\|^2$ es una función κ_∞, $\eta\,\|\gamma(k)\|^2$ es una función κ. Así que $L(k)$ admite una función ISS-lyapunov como en la definición 2. Por el teorema 1, la dinámica del error de identificación es estable para entradas estables. Se conoce $L(k)$ como función de $e(k)$ y $\gamma(k)$. La entrada corresponde al segundo término de (3.71), es decir, al error de modelado $\gamma(k)$. Los estados corresponden al primer término de (3.70), es decir, el error de identificación $e(k)$. Porque las entradas $\gamma(k)$ son acotadas y la dinámica es ISS, los estados $e(k)$ son acotados. (3.70) puede ser reescrito como

$$\begin{aligned}\Delta L(k) &= -\eta\frac{\|e(k)\|^2}{\left[1+\max_k\left(\|\varphi[s(k)]\|^2\right)\right]^2} + \eta\,\|\gamma(k)\|^2 \\ &\leq -\eta\frac{\|e(k)\|^2}{\left[1+\max_k\left(\|\varphi[s(k)]\|^2\right)\right]^2} + \eta\bar{\gamma}\end{aligned}$$

Así

$$\Delta L(k) \leq -\pi e_k^2 + \eta\gamma_k^2 \leq \pi e_k^2 + \bar{\gamma} \tag{3.72}$$

Sumando (3.72) de 1 hasta T, y utilizando $L_k > 0$ y L_1 es una constante, se obtiene

$$\begin{aligned}L_T - L_1 &\leq -\eta\sum_{k=1}^{T}\|e_N(k)\|^2 + T\eta\bar{\gamma} \\ \eta\sum_{k=1}^{T}\|e_N(k)\|^2 &\leq L_1 - L_T + T\eta\bar{\mu} \leq L_1 + T\eta\bar{\gamma}\end{aligned}$$

por lo que (3.69) es estable. ■

3.3.3. Redes jerárquicas recurrentes FCMAC

Para la identificación de sistemas no lineales mediante redes jerárquicas recurrenntes, solo se empleo la recurrencia global. La salida de una red recurrente CMAC difusa puede ser expresada en notación vectorial como:

$$\begin{aligned}
&\beta\hat{x}\left(k+1\right)=\sum_{i=1}^{l}w_{1,i}\varphi_{1,i}\left[s\left(k\right)\right]+\sum_{i=1}^{l}w_{2,i}\varphi_{2,i}\left[s\left(k\right)\right]u\left(k\right)\\
&\text{o }\beta\hat{x}\left(k+1\right)=W_1^T\varphi_1\left[s\left(k\right)\right]+W_2^T\varphi_2\left[s\left(k\right)\right]u\left(k\right)\\
&\widehat{y}(k)=\hat{x}\left(k+1\right)
\end{aligned}\tag{3.73}$$

Donde w_k desempeña el papel en la conectividad de los pesos de la red, W_j $(j=1,2)$ son los valores de los pesos ajustables en forma matricial, $\varphi_j\left(s\right)$ es la función base definida como $\varphi_p=\prod_{i=1}^{n}\lambda_p\left(\mu_{A_j^i,}\right)/\sum_{p=1}^{l}\prod_{i=1}^{n}\lambda_p\left(\mu_{A_j^i,}\right)$. Se realiza l $(p=1\cdots l)$ veces la asociación del vector de entrada $\hat{x}=[\hat{x}_1,\cdots,\hat{x}_n]\in\Re^n$ a una salida lingüistica y. Cada variable de entrada $s_i\left(i=1\ldots n\right)$ presenta m quantizaciones, el espacio de memoria requerido es de $l\times m^n$. La cantidad de memoria se incrementa exponencialmente al aumentar el número de variables de entrada, el cuál representa un serio problema en aplicaciones de redes CMAC difusas donde existe el problema de usar enormes cantidades de memoria y es indispensable optimizar su uso. Un método para superar esta dificultad es utilizar una estructura jerárquica con una red recurrente CMAC difusa, este tipo de sistema presenta una propiedad interesante, ya que el número de memoria requerido para construir el sistema jerárquico se incrementa sólo linealmente con el número de variables de entrada. Por ejemplo cada sub-bloque tiene dos entradas, el espacio de memoria es $l\times m^2\times b$, donde b es el número del sub-bloque, una red jerárquica recurrente CMAC difusa (HRFCMAC) con tres bloques (RFCMAC_1, RFCMAC_2, RFCMAC_3), se presenta en la figura Fig.3.10, donde $u_{3,1}=\hat{y}_1$, $u_{3,2}=\hat{y}_2$. Para RFCMAC_3 se tiene:

$$\begin{aligned}
\beta\hat{x}_3\left(k+1\right)=&\sum_{i=1}^{l}w_{3,1,i}\varphi_{3,1,i}\left[\hat{x}_3\left(k\right)\right]\\
&+\sum_{i=1}^{l}w_{3,2,i}\varphi_{3,2,i}\left[\hat{x}_3\left(k\right)\right]u_{3,i}\left(k\right)
\end{aligned}$$

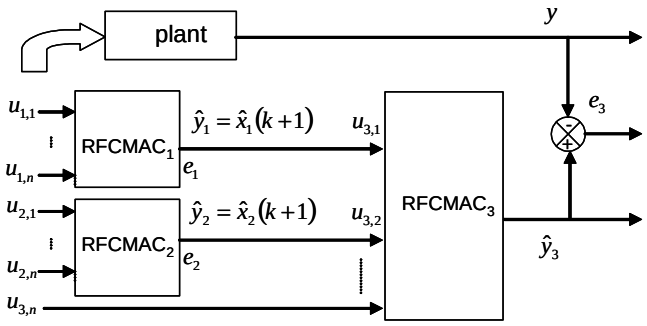

Figura 3.10: Red jerárquica recurrente FCMAC.

Identificación de sistemas via HRFCMAC

Se realiza el siguiente ejemplo para explicar como utilizar la técnica de retro-propagación (backpropagation) para las redes neuronales jerárquicas recurrentes CMAC difusas. La salida de cada bloque jerárquico de la CMAC difusa está dado por:

$$\widehat{x}_r(k+1) = \hat{y}_r = \sum_{i=1}^{l} w_{r,1,i}\varphi_{r,1,i}[x_r(k)] + \sum_{i=1}^{l} w_{r,2,i}\varphi_{r,2,i}[x_r(k)]\,u(k)\,; r = 1,2,3 \qquad (3.74)$$

La salida de cada bloque jerárquico es utilizado en el siguiente nivel jerárquico como variables lingüisticas de entrada. Sin embargo, las salidas de los bloques en las capas intermedias de una estructura jerárquica en la mayoría de los casos, es de naturaleza artificial, es decir, no posee ningún sentido físico. Si estas salidas intermedias son usadas como variables de entrada a la siguiente capa, entonces la reglas difusas involucradas en los bloques intermedios de la estructura jerárquica no poseen significado físico por lo que resultan más difíciles de diseñar. El índice de rendimiento se define como: $J = \frac{1}{2}e_3^2$, $e_3 = \widehat{y}_3 - y$. Para RFCMAC$_3$, El algoritmo de aprendizaje es:

$$\begin{aligned} w_{3,1,i}(k+1) &= w_{3,1,i}(k) - \eta\varphi_{3,1,i}(k)\,e_3(k) \\ w_{3,2,i}(k+1) &= w_{3,2,i}(k) - \eta\varphi_{3,2,i}(k)\,e_3(k)\,u(k) \end{aligned} \qquad (3.75)$$

Donde $\eta > 0$ es la taza de aprendizaje. Para el subsistema RFCMAC$_2$, si se desea actualizar $w_{2,i}^2$, Se calcula $\frac{\partial J}{\partial w_{3,2,i}} = \frac{\partial J}{\partial \widehat{y}_3}\frac{\partial \widehat{y}_3}{\partial \widehat{y}_2}\frac{\partial \widehat{y}_2}{\partial w_{2,2,i}}$. De la figura 3.10 se conoce $\frac{\partial \widehat{y}_3}{\partial \widehat{y}_2}$ que corresponde a

$x_{2,1,2}(k)$, así que:

$$\begin{aligned}
&\frac{\partial J}{\partial \widehat{y}_3} = \widehat{y}_3 - y = e_3(k) \\
&\frac{\partial \widehat{y}_3}{\partial \widehat{y}_2} = \frac{\partial \widehat{y}_3}{\partial z_{3,1,i}} \frac{\partial z_{3,1,i}}{\partial \widehat{y}_2} \\
&= \left[\frac{a_3}{b_3^2} - \frac{w_{3,1,i}}{b_3} \right] z_{3,h} \left[2 \frac{\widehat{y}_2 - c_{3,2}^i}{\left(\sigma_{3,2}^i\right)^2} \right] \\
&\frac{\partial \widehat{y}_2}{\partial w_{2,1,i}} = \varphi_{2,1,i} = \frac{z_{2,1,i}}{b_2}
\end{aligned} \tag{3.76}$$

Donde $z_{3,1,i} = \alpha_k = \prod_{l=1}^{n} \lambda_i \left(\mu_{A_j^l} \right)$, $a_{3,1} = \sum_{h=1}^{l} w_{3,1,i} z_{3,1,i}$, $b_3 = \sum_{h=1}^{l} z_{3,1,i}$. El método de aprendizaje del gradiente para $w_{2,1,i}$ es

$$\begin{aligned}
&w_{2,1,i}(k+1) = w_{2,1,i}(k) \\
&-\eta \varphi_{2,1,i}(k) \frac{z_{2,1,i}}{b_2} 2 \frac{\widehat{y}_3 - w_{3,1,i}}{b_3} z_{3,1,i} \frac{\widehat{y}_2 - c_{3,2}^i}{\left(\sigma_{3,2}^i\right)^2} e_3(k) \\
&= w_{2,1,i}(k) - \eta \varphi_{2,1,i}(k) e_2(k)
\end{aligned} \tag{3.77}$$

Donde

$$\begin{aligned}
e_2(k) &= 2 \frac{\widehat{y}_3 - w_{3,1,i}}{b_3} z_{3,1,i} \frac{\widehat{y}_2 - c_{3,2}^i}{\left(\sigma_{3,2}^i\right)^2} e_3(k) \\
&= \frac{\partial \widehat{y}_3}{\partial \widehat{y}_2} e_3(k)
\end{aligned}$$

Por lo que (3.77) presenta la misma forma como (3.75). De manera similar se puede obtener con

$$\begin{aligned}
&e_1(k) = \frac{\partial \widehat{y}_3}{\partial \widehat{y}_1} e_3(k), \\
&\frac{\partial \widehat{y}_3}{\partial \widehat{y}_1} = 2 \frac{\widehat{y}_3 - w_{3,1,i}}{b_3} z_{3,1,i} \frac{y_2 - c_{3,1}^i}{\left(\sigma_{3,1}^i\right)^2} \\
&w_{1,1,i}(k+1) = w_{1,1,i}(k) - \eta \varphi_{1,1,i}(k) e_1(k)
\end{aligned}$$

Para el caso general, como se muestra en la Fig. 3.11. El procedimiento de entrenamiento se explica a continuación:

1) De acuerdo a la estructura de la red HRFCMAC, se calcula la salida de cada bloque de la red neuronal RFCMAC mediante (3.73). Algunas de las salidas de los bloques de la red neuronal jerárquica recurrente CMAC difusa pueden ser las entradas a los bloques del siguiente nivel.

2) Se calcula el error para cada bloque. Se inicia con el último bloque, el error de identificación es $e_o(t) = \widehat{y}_o(t) - y(t)$, donde $e_o(k)$ es el error de identificación, $\widehat{y}_o(k)$ es la salida del

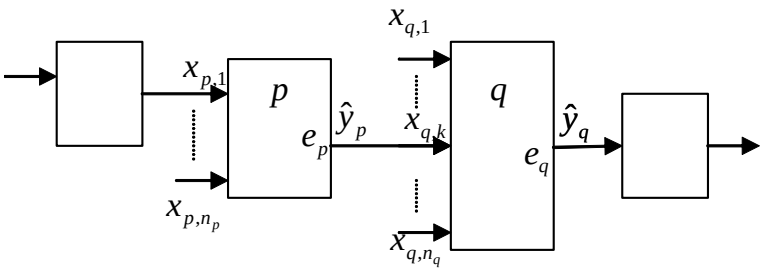

Figura 3.11: Red HRFCMAC.

sistema jerárquico recurrente CMAC difuso, $y(k)$ es la salida de la planta. Entonces se utiliza la retro-propagación para minimizar el error en cada bloque de la estructura jerárquica de la red CMAC, ver la figura 3.11. Se calcula el error para el bloque p (definido como e_p), de la misma forma hasta el bloque q (definido como e_q). Por la regla de la cadena discutida anteriormente:

$$e_p(k) = 2\frac{\widehat{y}_q - w_{i,q}}{b_q} z_{q,h} \frac{\widehat{y}_p - c_{i,q,p}}{\left(\sigma^i_{q,p}\right)^2} e_q(k) \tag{3.78}$$

3) Entrenamiento de la función Gaussiana (función de membresia en la premisa y en la parte consecuente) para cada bloque independientemente, para el $p-ésimo$ bloque el algoritmo de retro-propagación (backpropagation) esta dado por:

$$w_{p,h}(k+1) = w_{p,h}(k) - \eta \varphi_{p,h}(t)\, e_p(k) \tag{3.79}$$

Aprendizaje estable para cada bloque

Se asume que la función base $\varphi_{r,h}$ de la red CMAC para cada bloque es conocido, solamente los pesos necesitan ser actualizados para la identificación de sistemas. Se diseña un algoritmo de aprendizaje estable tal que la salida $\widehat{y}(k)$ de la red FCMAC (3.73) pueda identificar la salida $y(k)$ de la planta no lineal (3.1). Se define el error de identificación $e(k)$ como:

$$e(k) = \widehat{y}(k) - y(k) \tag{3.80}$$

Por (3.78), $e(k)$ puede ser propagado para cada sub-bloque $e_p(k)$. Se define una salida virtual de la planta como $y_p(k)$, el cuál corresponde a la salida del sub-bloque $\widehat{y}_p(k)$, tal que $e_p(k) = \widehat{y}_p(k) - y_p(k)$, expresa el error para el $p - ésimo$ bloque de la planta no lineal. De acuerdo a la teoría de aproximación de lógica difusa y redes neuronales [27], el proceso no lineal a ser identificado (3.4), puede ser representado como:

$$\begin{aligned} &\beta\hat{x}(k+1) = A\hat{x}(k) + W_1^{*T}\varphi_1[\hat{x}(k)] \\ &+W_2^{*T}\varphi_2[\hat{x}(k)]U(k) + \nu(k) \\ &y(k) = \hat{x}(k+1) \end{aligned} \tag{3.81}$$

Donde W_1^* y W_2^* son los pesos desconocidos los cuáles pueden minimizar las dinámicas no modeladas $\nu(k)$. El error de identificación puede ser representado por (3.73) y (3.81),

$$\begin{aligned} \beta e_p(k+1) = Ae_p(k) + \widetilde{W}_1(k)\varphi_1[\hat{x}(k)] \\ +\widetilde{W}_2^T\varphi_2[\hat{x}(k)]u(k) - \nu(k) \end{aligned} \tag{3.82}$$

Donde $\widetilde{W}_1(k) = W_1(k) - W_1^*$, $\widetilde{W}_2(k) = W_2(k) - W_2^*$. En este trabajo solo estamos interesados en la identificación en lazo abierto, Se asume que la planta (3.1) presenta entradas acotadas - salidas acotadas (BIBO estable), *i.e.*, $u(k)$ y $y(k)$ son acotadas. Por los límites de la función base φ, $\nu(k)$ en (3.81) es acotado. El siguiente teorema presenta el algoritmo estable del gradiente descendente para modelado neuro-difuso, el cuál es una aportación del trabajo de tesis.

Teorema 3.5 *Si la red neuronal recurrente CMAC difusa (3.73) es usado para identificar una planta no lineal (3.1) y los eigenvalores de A se seleccionan como* $-1 < \lambda(A) < 0$*, La siguiente ley de actualización del gradiente sin modificación robusta puede hacer el error de identificación* $e(k)$ *acotado (estable en el sentido* L_∞*).*

$$\begin{aligned} W_1(k+1) = W_1(k) - \eta(k)\varphi_1[x(k)]e^T(k) \\ W_2(k+1) = W_2(k) - \eta(k)\varphi_2[x(k)]u(k)e^T(k) \end{aligned} \tag{3.83}$$

Donde $\eta(k)$ *satisface*

$$\eta(k) = \begin{cases} \dfrac{\eta}{1 + \|\varphi_1\|^2 + \|\varphi_2 u\|^2} & si\ \beta \|e(k+1)\| \geq \|e(k)\| \\ 0 & si\ \beta \|e(k+1)\| < \|e(k)\| \end{cases}$$

$0 < \eta \leq 1$.

Demostración. Se seleciona una función candidata de Lyapunov como:

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

donde $\left\|\widetilde{W}_1(k)\right\|^2 = \sum_{i=1}^{n} \widetilde{w}_1(k)^2 = tr\left\{\widetilde{W}_1^T(k)\,\widetilde{W}_1(k)\right\}$. De la ley de actualización (3.83) se tiene:

$$\widetilde{W}_1(k+1) = \widetilde{W}_1(k) - \eta(k)\,\varphi_1[x(k)]\,e^T(k)$$

So

$$\begin{aligned} &\Delta V(k) = V(k+1) - V(k) \\ &= \left\|\widetilde{W}_1(k) - \eta(k)\,\varphi_1 e(k)^T\right\|^2 - \left\|\widetilde{W}_1(k)\right\|^2 \\ &+ \left\|\widetilde{W}_2(k) - \eta(k)\,\varphi_2 u(k)\,e^T(k)\right\|^2 - \left\|\widetilde{W}_2(k)\right\|^2 \\ &= \eta^2(k)\,\|e(k)\|^2\,\|\varphi_1\|^2 - 2\eta(k)\left\|\varphi_1\widetilde{W}_1(k)\,e^T(k)\right\| \\ &+\eta^2(k)\,\|e(k)\|^2\,\|\varphi_2 u(k)\|^2 - 2\eta(k)\left\|\varphi_2 u(k)\,\widetilde{W}_2(k)\,e^T(k)\right\| \end{aligned}$$

Existe una constante $\beta > 0$, tal que si:

$$\|\beta e(k+1)\| \geq \|e(k)\|$$

utilizando (3.82) y $\eta(k) \geq 0$

$$\begin{aligned} &-2\eta(k)\left\|\varphi_1\widetilde{W}_1(k)\,e^T(k)\right\| - 2\eta(k)\left\|\varphi_2 u(k)\,\widetilde{W}_2(k)\,e^T(k)\right\| \\ &\leq -2\eta(k)\left\|e^T(k)\right\|\|\beta e(k+1) - Ae(k) - \nu(k)\| \\ &= -2\eta(k)\left\|\begin{array}{c} e^T(k)\,\beta e(k+1) \\ -e^T(k)\,Ae(k) - e^T(k)\,\nu(k) \end{array}\right\| \\ &\leq -2\eta(k)\left\|e^T(k)\,\beta e(k+1)\right\| \\ &+2\eta(k)\,e^T(k)\,Ae(k) + 2\eta(k)\left\|e^T(k)\,\nu(k)\right\| \\ &\leq -2\eta(k)\,\|e(k)\|^2 + 2\eta(k)\,\lambda_{\text{máx}}(A)\,\|e(k)\|^2 \\ &+\eta(k)\,\|e(k)\|^2 + \eta(k)\,\|\nu(k)\|^2 \end{aligned}$$

donde $0 < \eta \leq 1$

$$\begin{array}{l} \Delta V(k) \leq \eta^2(k) \|e(k)\|^2 \|\varphi_1\|^2 + \eta^2(k) \|e(k)\|^2 \|\varphi_2 u(k)\|^2 \\ -\eta(k) \|e(k)\|^2 + 2\eta(k) \lambda_{\text{máx}}(A) \|e(k)\|^2 + \eta(k) \|\nu(k)\|^2 \\ = -\eta(k) \begin{bmatrix} (1 - 2\lambda_{\text{máx}}(A)) \\ -\eta \dfrac{\|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2}{1 + \|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2} \end{bmatrix} e^2(k) \\ +\eta_k^2 \nu^2(k) \leq -\pi e^2(k) + \eta \nu^2(k) \end{array} \tag{3.84}$$

donde:

$$\pi = \frac{\eta}{1+\kappa} \left[1 - 2\lambda_{\text{máx}}(A) - \frac{\kappa}{1+\kappa} \right]$$

y $\kappa = \underset{k}{\text{máx}} \left(\|\varphi_1\|^2 + \|\varphi_2 u(k)\|^2 \right)$. Por lo que $-1 < \lambda(A) < 0$, $\pi > 0$

$$n \,\text{mín}\left(\widetilde{w}_i^2\right) \leq V(k) \leq n \,\text{máx}\left(\widetilde{w}_i^2\right)$$

donde $n \times \text{mín}(\widetilde{w}_i^2)$ y $n \times \text{máx}(\widetilde{w}_i^2)$ son funciones de clase $\mathcal{K}_\infty$, y $\pi e^2(k)$ es una función clase $\mathcal{K}_\infty$, $\eta\nu^2(k)$ es una función de clase $\mathcal{K}$, tal que $V(k)$ admite una función suave de Lyapunov como en la definición 2. Del Teorema 1, la dinámica del error de identificación es entrada-estado estable. La entrada corresponde al segundo término de la última linea de(3.84), *i.e.*, el error de modelado $\nu(k)$, los estados corresponden al primer término de la última linea de (3.84), *i.e.*, el error de identificación $e(k)$. Por que la entrada $\nu(k)$ es acotada y la dinamica es ISS, los estados $e(k)$ son acotados. ■

Comentario 3.3 *La condicion "$\eta(k) = 0$ si $\beta \|e(k+1)\| < \|e(k)\|$" es la zona muerta. si β se elije con un valor muy grande, la zona muerta llega a ser pequeña.*

3.4. Simulaciones

Para poder realizar un análisis comparativo entre las redes FCMAC con recurrencia local, global y la combinación de ambas, se considera el siguiente sistema no lineal propuesto por [16] y [26], para ilustrar los resultados de la identificación.

$$\begin{aligned} x_1(k+1) &= x_2(k), \\ x_2(k+1) &= x_3(k) \\ x_3(k+1) &= \frac{x_1(k)x_2(k)x_3(k)u(k)[x_3(k)-1]+u(k)}{1+x_2(k)^2+x_3(k)^2} \end{aligned} \tag{3.85}$$

Se define el vector de estados como $y(k) = [x_1(k), x_2(k), x_3(k)]^T$, la señal de entrada se selecciona como en [16] y [26].

$$u(k) = \begin{cases} \sin(\frac{\pi}{25}k) & 0 < k < 250 \\ 1{,}0 & 250 \leq k < 500 \\ -1{,}0 & 500 \leq k < 750 \\ 0{,}3\sin(\frac{\pi}{25}k) + 0{,}1\sin(\frac{\pi}{32}k) + 0{,}6\sin(\frac{\pi}{10}k) & 750 \leq k < 1000 \end{cases} \tag{3.86}$$

A la red neuronal con recurrencia local la denotamos como LRFCMAC, con recurrencia global la denotamos como GRFCMAC y a la red que presenta la combinación de ambas recurrencias local y global la denotamos como GLRFCMAC. En la figura 3.12 se comparan los resultados de la identificación de las diferentes topologías de redes FCMAC desarrollados en la tesis con el trabajo de redes neuronales FCMAC convencionales presentados por [7]. Se realiza la simulación con los siguientes datos $h = 5$ (número de hipercubos), $n_a = 4$ (numero de unidades de asociación), la entrada $s = 3$, la salida $y = 1$, $l = 8$. La siguiente tabla muestra la cantidad de operaciones realizadas por cada una de las capas de las diferentes topologías de redes FCMAC.

Capa	FCMAC	LRFCMAC	GRFCMAC	GLRFCMAC
L1: Entrada	n	n	$n+p$	$n+p$
L2: Funciones de membresia	$n \times l$	$n \times l + n \times l$	$n \times l$	$n \times l + n \times l$
L3: Antecedente	m^n	m^n	m^n	m^n
L4: Consecuente	m^n	m^n	m^n	m^n
L5: Salida	p	p	p	p

Tabla 1. Número de operaciones realizadas en cada capa de la red neuronal.

Las variables de la Tabla 1 son: n es el número de entradas, p el número de salidas, l el número de funciones de membresia por cada entrada y m^n es el número de reglas de inferencia. La Tabla 1 muestra que la red GLRFCMAC realiza el mayor número de operaciones, principalmente en la capa $L2$, sin embargo como se observa en la figura 3.13, el error de identificación que presenta es menor que con las redes neuronales FCMAC, LRFCMAC, GRFCMAC.

3.4.1. Red HFCMAC vs HRFCMAC

Uno de los problemas que presenta la red FCMAC convencional es que si las entradas son mayores a 3, la memoria que soporta la topología de la red se incrementa, haciendolo difícil de implementar, este problema se conoce como ¨el problema de la dimensionalidad¨. En los sitemas difusos para resolver este problema se propuso una estructura jerárquica [23], [24], [27], para que las reglas de inferencia creciera linealmente y no exponencialmente. Este mismo principio se aplica a las redes FCMAC con estructura jerárquica y con estructura jerárquica recurrente. Para realizar la identificación de un sistema no lineal mediante redes FCMAC con estructura jerárquica se utiliza la planta dada por (3.85). La entrada al primer bloque es $[x_1(k), x_2(k)]$, la entrada al segundo bloque es $[x_3(k), u(k)]$,con $n = 3, m = 5, na = 10$, se utilizan 1000 datos de entrenamiento. Los resultados de la identificación se muestra en la figura 3.14.

Como se puede observar en las figuras 3.14 y 3.15, ambas topologias FCMAC con estructura jerárquica pueden identificar un sistema no lineal, resuelven el problema del crecimiento de la memoria y además pueden ser utilizados en sistemas dinámicos.

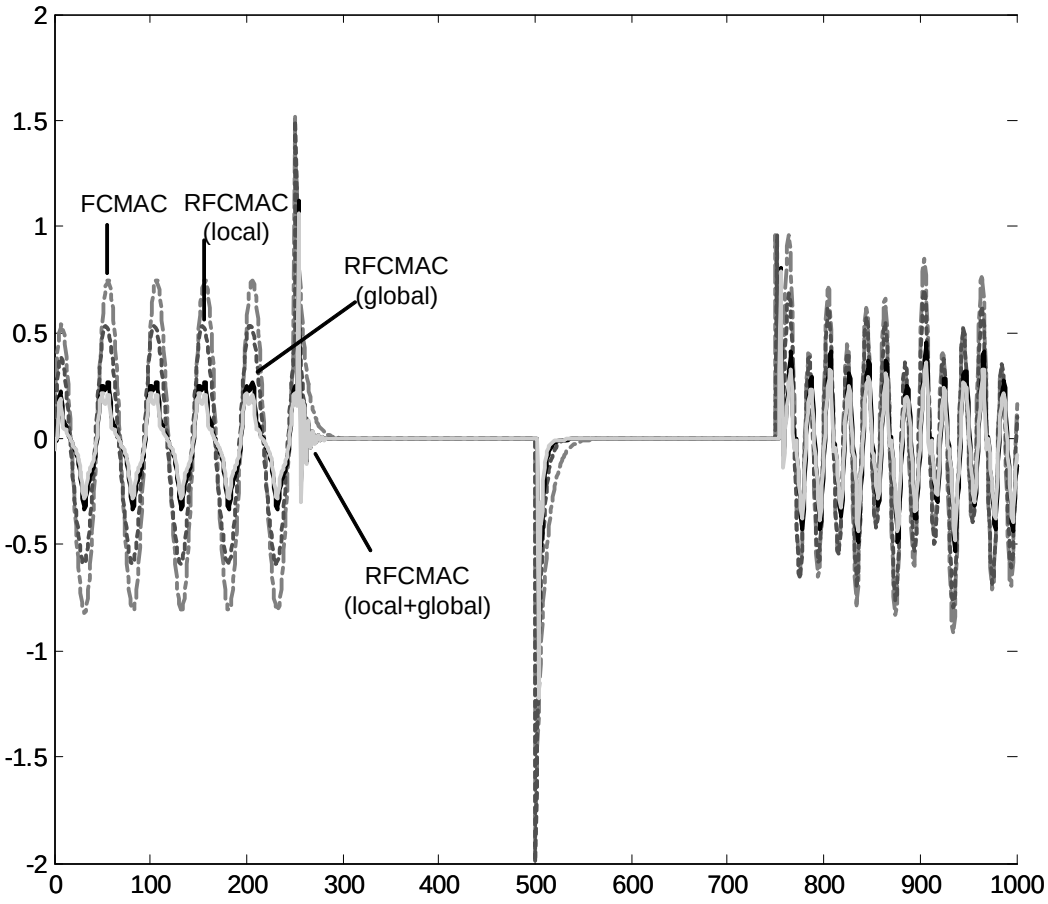

Figura 3.12: Comparacion de resultados de las redes FCMAC.

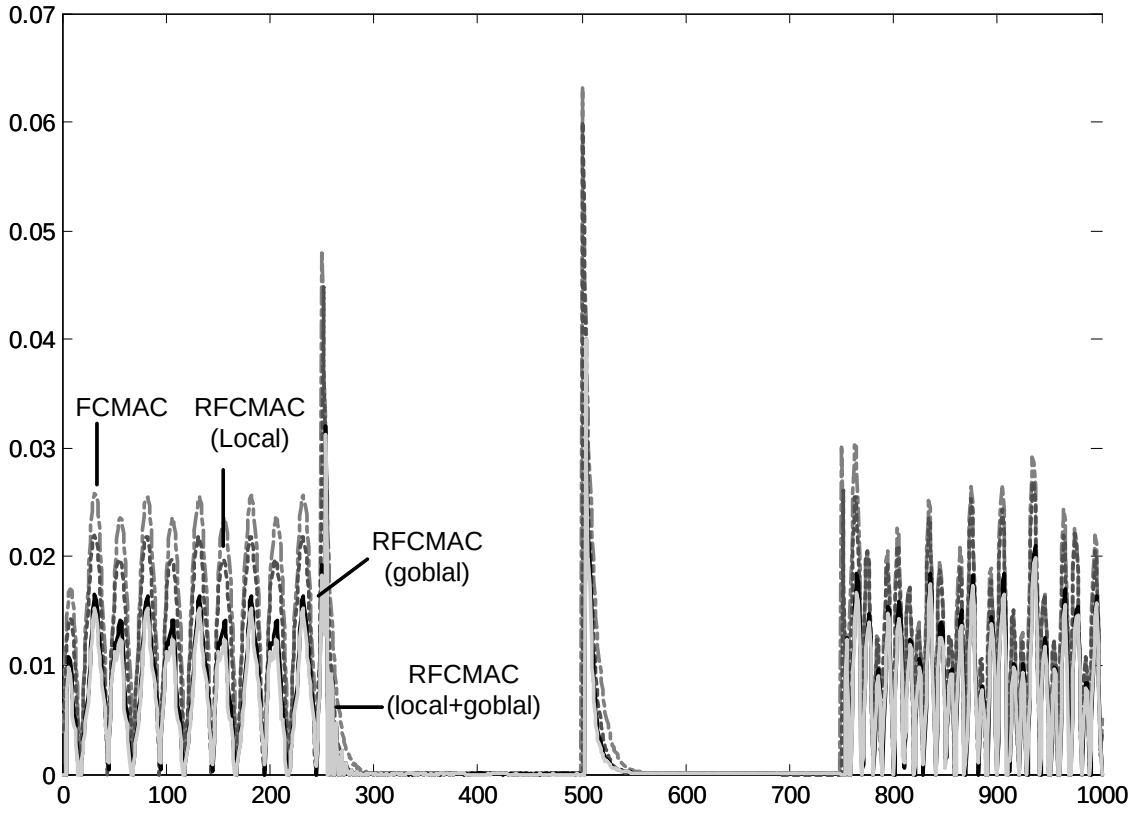

Figura 3.13: Error Cuadrático de las redes recurrentes FCMAC

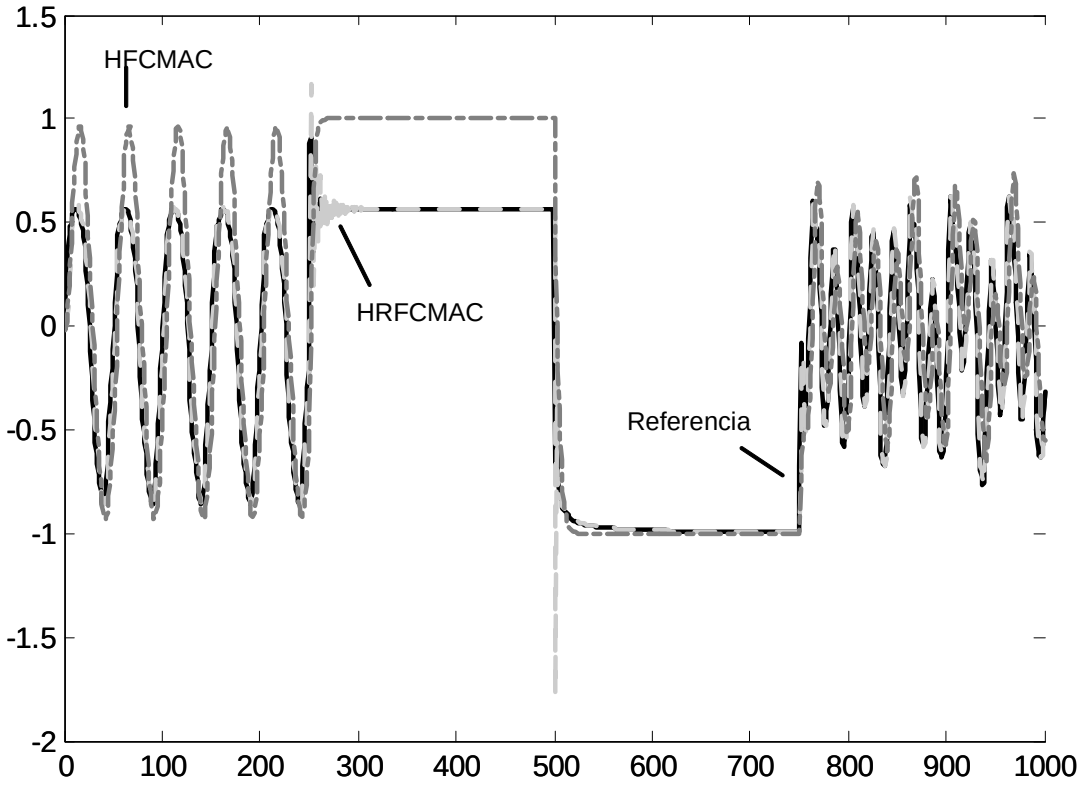

Figura 3.14: Identificación mediante la red HFCMAC vs HRFCMAC.

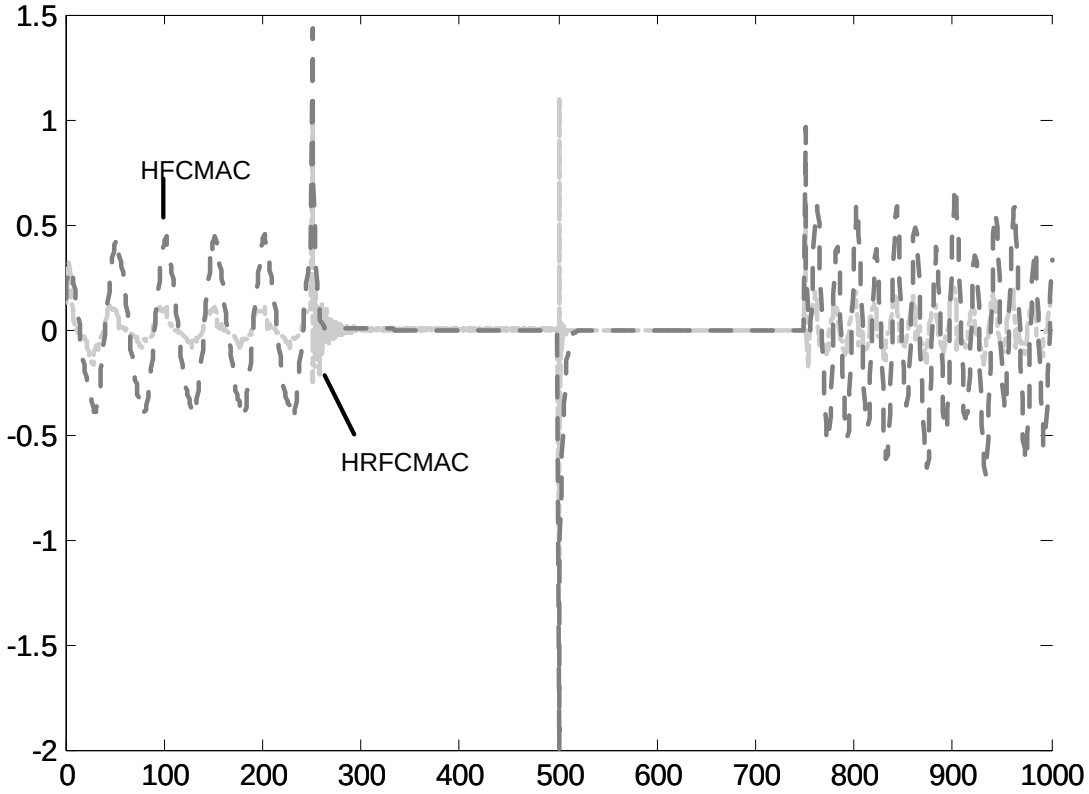

Figura 3.15: Error de Identificación mediante la red HFCMAC y la HRFCMAC.

Capítulo 4

Control de sistemas dinámicos

Un controlador puede ser usado en un sistema en lazo abierto ó en un sistema a lazo cerrado. Los sistemas en lazo en lazo cerrado pueden rechazar las perturbaciones y realizar un control más exacto, una de las desventajas es que la retroalimentación puede causar problemas en la estabilidad del sistema, los cuáles no están presentes en los sistemas a lazo abierto.

4.1. Métodos de Control Adaptable

Las técnicas de control convencional que presentan retroalimentación están basados en análisis en el dominio del tiempo y en el dominio de la frecuencia, donde los procesos y los controladores están descritos por funciones de transferencia. Cuando un sistema $f(\cdot)$ es lineal y sus parámetros son conocidos, las técnicas de diseño de controladores convencionales pueden ser utilizados para encontrar un controlador conveniente para dicho sistema, teniendo una relación entrada-salida deseada, es decir, una función de transferencia en lazo cerrado.

El comportamiento del control en lazo cerrado está definido en el dominio de la frecuencia por los parámetros de margen de fase, ancho de banda ó en el dominio del tiempo por la respuesta transitoria, el error en el estado estacionario, etc. El proposito del diseño del controlador es hacer el error cero en estado estacionario, una rápida respuesta la cuál está

relacionada con el ancho de banda y que se mantenga la estabilidad, lo cuál está relacionado con el margen de fase.

Un controlador muy usado en lazo cerrado es el control Proporcional, Integral, Derivativo (PID). La tarea de diseño se reduce a ajustar las ganancias K_p, K_i, K_d de los controladores (sintonización Ziegler-Nichols).

Al sufrir cambios un sistema de control no lineal (tales como su punto de operación) sufre variaciones en sus parámetros, estas variaciones pueden afectar severamente la estabilidad y el comportamiento del sistema. El uso del control adaptable tiene como objetivo rediseñar automáticamente el controlador del sistema cuando ocurren estos cambios en los parámetros. Existen dos esquemas de control adaptable:

- Control adaptable con modelo de referencia (MRAC):
- El sistema de control adaptable con modelo de referencia directo, modifica los parámetros del controlador directamente.

Un sistema de control adaptable con modelo de referencia indirecto, modifica los parámetros del modelo de referencia, y los parámetros de este modelo de referencia son utilizados para calcular los parámetros del controlador.

- Control auto-sintonizado: Un regulador auto-sintonizado identifica los parámetros de la planta cuando el sistema esta corriendo. Los parámetros identificados son utilizados para rediseñar el controlador.

En un control adaptable los parámetros se encuentran disponibles y existen mecanismos para ajustarlos en línea, basado en las señales del sistema. Convencionalmente los controladores adaptables pueden ser clasificados de acuerdo a la manera por la cuál sus parámetros son ajustados. Los métodos de ajuste se clasifican en dos categorias: Control adaptable indirecto y control adaptable directo. En el control directo, los parámetros del controlador son directamente ajustados para reducir la norma de la salida del error dado por la salida de la planta y la trayectoria de referencia deseada. En el control indirecto, los parámetros de la

planta con estimados y el controlador es diseñado asumiendo que los parámetros de la planta estimada representan los valores verdaderos de la planta original.

4.2. Control adaptable basado en las redes CMAC

Considere el siguiente sistema dinámico no lineal en tiempo continúo de $n-ésimo$ orden, libre de perturbaciones externas:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= x_3(t) \\ &\vdots \\ \dot{x}_n(t) &= f(x_1, x_2, ..., x_n) + g(x_1, x_2, ..., x_n)\, u \\ y &= x_1 \end{aligned} \tag{4.1}$$

ó escrita en su forma equivalente:

$$\begin{aligned} x^n &= f\left(x, \dot{x}, ..., x^{n-1}\right) + g\left(x, \dot{x}, ..., x^{n-1}\right) u \\ y &= x \end{aligned} \tag{4.2}$$

donde f y g son funciones acotadas, $x \in R^n$ es el vector de estados, $u \in R$ es la entrada de control, $y \in R$ es la salida de control del sistema. Solo se consideran los sistemas no lineales que pueden ser representados por (4.1) ó (4.2). La ecuación (4.2) puede reescribirse en su representación en espacio de estados como:

$$\begin{aligned} \dot{x}(t) &= Ax + B\left[f(x) + g(x)\, u\right] \\ y &= C^T x \end{aligned} \tag{4.3}$$

Se define A, B y C como:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

y $x = [x_1, x_2, ..., x_n]^T = \left[x, \dot{x}, ..., x^{(n-1)}\right]^T \in R^n$, es el vector de estados, se da por hecho que todos los estados x_i y la salida y, están disponibles para ser medibles. Para que el sistema (4.2) sea controlable, se requiere que $g(x) \neq 0$ para x en una región de controlabilidad $U_c \subset R^n$. Sin perdida de generalidad, se da por hecho que $0 < g(x) < \infty$, para $x \in U_c$.

Si $f(x)$ y $g(x)$ son conocidos, entonces se puede elejir una ley de control u para cancelar las no linealidades y diseñar el controlador basado solamente en la teoría de control lineal. Se define el vector de la señal de referencia como y_d, sea $e = y_d - x = y_d - y \in R$, el error del vector de trayectoria e se define como:

$$\begin{aligned} y_d &= \left[y_d, ..., y_d^{n-1}\right]^T \in R^n \\ e &= y_d - y \\ e &= \left[e, \dot{e}, ..., e^{n-1}\right]^T = [e_1, e_2, ..., e_n]^T \in R^n \end{aligned} \tag{4.4}$$

El objetivo del control es forzar la salida del sistema y, a seguir una señal de referencia acotada y_d, bajo la condición de que todas las señales involucradas son acotadas. Sea $k = [k_1, ..., k_n]^T \in R^n$ tal que todas las raíces del polinomio $s^n + k_1 s^{n-1} + ... + k_n$, se encuentren del lado izquierdo del plano complejo. Si las funciones $f(x)$ y $g(x)$ son conocidos, entonces se puede elejir una ley de control u^* ideal como:

$$u^* = \frac{1}{g(x)} \left[-f(x) + k^T x + y_d^n\right] \tag{4.5}$$

Donde $u = u^*$, substituyendo (4.5) en (4.1), se obtiene el sistema en lazo cerrado dado por la expresión:

$$e^n + k_n e^{n-1} + ... + k_1 e = 0$$

Se define:

$$\Lambda = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ -k_n & -k_{n-1} & -k_{n-2} & -k_{n-3} & \cdots & -k_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{4.6}$$

El sistema en lazo cerrado se reescribe como:

$$\dot{e}(t) = \Lambda e \tag{4.7}$$

Donde el objetivo de control implica iniciar de cualquier condición inicial, se tiene $lim_{t\to\infty} e(t) = 0$, tal que Λ es estable, la salida de la planta y, converge a la salida ideal y_d, asintóticamente.

4.2.1. Control adaptable indirecto basado en redes CMAC

Si $f(x)$ y $g(x)$ son funciones desconocidas, el control ideal (4.5), no puede ser implementado, pues al no conocerse dichas funciones, no se pueden cancelar las no linealidades. En este trabajo se proponen dos controladores HFCMAC llamados: $\hat{f}(x \mid w_f)$ y $\hat{g}(x \mid w_g)$, para aproximar las funciones desconocidas $f(x)$ y $g(x)$ respectivamente, esto es, un controlador indirecto es diseñado para minimizar el error del esquema de control equivalente, el cuál está dado por:

$$u_I = \frac{1}{\hat{g}(x \mid w_g)} \left[-\hat{f}(x \mid w_f) + k^T x + y_d^n \right] \tag{4.8}$$

donde $\hat{f}(x \mid w_f)$ y $\hat{g}(x \mid w_g)$ se obtienen de la salida de las redes neuronales HFCMAC, las cuáles están dadas por:

$$\begin{aligned} \hat{f}\left(x \mid w_f\right) &= w_f^T \varphi_f\left(x\right) \\ \hat{g}\left(x \mid w_g\right) &= w_g^T \varphi_g\left(x\right) \end{aligned} \tag{4.9}$$

Donde w_f y w_g son los parámetros de los sistemas aproximados $\hat{f}$ y $\hat{g}$, respectivamente. Substituyendo (4.8) en (4.2) se obtiene la dinámica en lazo cerrado, del sistema de control HFCMAC dado por:

$$e^{(n)} = -k^T e + \left[\hat{f}\left(x \mid w_f\right) - f\left(x\right)\right] + \left[\hat{g}\left(x \mid w_g\right) - g\left(x\right)\right] u_I \tag{4.10}$$

La ecuación (4.10) se puede escribir en notación vectorial de la forma:

$$\dot{e}\left(t\right) = \Lambda e + B\left[\left[\hat{f}\left(x \mid w_f\right) - f\left(x\right)\right] + \left[\hat{g}\left(x \mid w_g\right) - g\left(x\right)\right] u_I\right] \tag{4.11}$$

donde Λ y B se definen como (4.6). Para el desarrollo de las leyes adaptables más convenientes para ajustar los pesos de la red HFCMAC, se elijen los parámetros óptimos w_f^* y w_g^* definidos por los siguientes términos:

$$\begin{aligned} w_f^* &= \arg \min_{w_f \in \Omega_f} \left(\sup_{x \in \Omega_x} \left| \hat{f}\left(x \mid w_f\right) - f\left(x\right) \right| \right) \\ w_g^* &= \arg \min_{w_g \in \Omega g} \left(\sup_{x \in \Omega_x} \left| \hat{g}\left(x \mid w_g\right) - g\left(x\right) \right| \right) \end{aligned} \tag{4.12}$$

Así $\hat{f}^*\left(x \mid w_f^*\right)$ y $\hat{g}^*\left(x \mid w_g^*\right)$ son los mejores aproximadores de $f\left(x\right)$ y $g\left(x\right)$. Donde $\Omega_f = \|w_f\| \leq U_f$, $\Omega_g = \|w_g\| \leq U_g$, son conjuntos compactos acotados, los cuáles corresponden a $w_f = w_{f_1}, ..., w_{f_n}$, $\quad w_g = w_{g_1}, ..., w_{g_n}$, y U_f, U_g son constantes positivas especificadas por el diseñador. Las salidas de las redes HFCMAC con sus respectivos pesos óptimos correspondientes $\hat{f}^*\left(x \mid w_f^*\right)$ y $\hat{g}^*\left(x \mid w_g^*\right)$ son utilizados para aproximar a las dos funciones desconocidas $f\left(x\right)$ y $g\left(x\right)$, tal que:

$$\begin{aligned} f(x) &= \hat{f}^*\left(x \mid w_f^*\right) + \epsilon_f \\ \epsilon_f &= \left| f(x) - \hat{f}^*\left(x \mid w_f^*\right) \right| \\ g(x) &= \hat{g}^*\left(x \mid w_g^*\right) + \epsilon_g \\ \epsilon_g &= \left| g(x) - \hat{g}^*\left(x \mid w_g^*\right) \right| \end{aligned}$$

donde ϵ_f y ϵ_g, son los errores de aproximación minimos de $\tilde{f}(x)$ y $\tilde{g}(x)$, los cuáles se definen a continuación:

$$\begin{aligned} \tilde{f} &= f(x) - \hat{f}(x \mid w_f) \\ &= \hat{f}^*\left(x \mid w_f^*\right) + \epsilon_f - \hat{f}(x \mid w_f) \\ &= w_f^* \varphi_f(x) + \epsilon_f - w_f^T \varphi_f(x) \\ &= \tilde{w}_f \varphi_f(x) + \epsilon_f \end{aligned} \tag{4.13}$$

$$\begin{aligned} \tilde{g} &= g(x) - \hat{g}(x \mid w_g) \\ &= \hat{g}^*\left(x \mid w_g^*\right) + \epsilon_g - \hat{g}(x \mid w_g) \\ &= w_g^* \varphi_g(x) + \epsilon_g - w_g^T \varphi_g(x) \\ &= \tilde{w}_g \varphi_g(x) + \epsilon_g \end{aligned} \tag{4.14}$$

donde $\tilde{w}_i^T \varphi_i = \tilde{w}_i^{*T} \varphi_{i-} w_i^T \varphi_i$, i denota a las funciones $f(x)$ y $g(x)$. En el proceso de diseño del controlador, el término de incertidumbre se añade mediante el siguiente término:

$$\begin{aligned} \xi &= \epsilon_f + \epsilon_g u_I \\ \xi &= \left| f(x) - \hat{f}^*\left(x \mid w_f^*\right) \right| + \left| g(x) - \hat{g}^*\left(x \mid w_g^*\right) \right| u_I \end{aligned} \tag{4.15}$$

Se realiza la consideración que la incertidumbre ξ está acotado por una constante positiva pequeña δ, $|\xi| \leq \delta$. Utilizando el procedimiento anterior, la ecuación (4.11) se reescribe como:

$$\dot{e}(t) = \Lambda e + B\left[\left[\hat{f}(x \mid w_f) - \hat{f}^*\left(x \mid w_f^*\right)\right] + \left[\hat{g}(x \mid w_g) - \hat{g}^*\left(x \mid w_g^*\right)\right] u_I + B\xi\right] \tag{4.16}$$

Substituyendo (4.9), en (4.16), se obtiene la siguiente ecuación dinámica en lazo cerrado, la cuál presenta una relación explicita entre el error de trayectoria e y los parámetros del controlador w_f y w_g:

$$\begin{aligned}
\dot{e}(t) &= \Lambda e + B\left[\left[w_f^T\varphi_f(x) - w_f^{*T}\varphi_f(x)\right] + \left[w_g^T\varphi_g(x) - w_g^{*T}\varphi_g(x)\right]u_I + \xi\right] \quad (4.17)\\
\dot{e}(t) &= \Lambda e + B\left[\left[w_f^T - w_f^{*T}\right]\varphi_f(x) + \left[w_g^T - w_g^{*T}\right]\varphi_g(x)\,u_I + \xi\right]\\
\dot{e}(t) &= \Lambda e + B\left[\tilde{w}_f^T\varphi_f(x) + \tilde{w}_g^T\varphi_g(x)\,u_I\right] + B\xi
\end{aligned}$$

4.2.2. Ley de aprendizaje para las redes neuronales HFCMAC

La función de la ley de aprendizaje es determinar un mecanismo de ajuste para w_f y w_g, tal que el error de trayectoria e, y los parámetros de error ϵ_f y ϵ_g, son minimizados. La meta al ajustar los parámetros es hacer $w_f^T \to w_f^{*T}$, y $w_g^T \to w_g^{*T}$, y por consecuencia, hacer $\hat{f}(x \mid w_f), \hat{g}(x \mid w_g)$ buenos estimadores de $f(x)$,$g(x)$, y $u_I \to u^*$.

$$\begin{aligned}
\hat{f}(x \mid w_f) &= w_{f_n}^T\varphi_{f_n}(x_n) \quad (4.18)\\
x_{n-1} &= w_{f_{n-1}}^T\varphi_{f_{n-1}}(x_{n-1})\\
&\vdots\\
x_1 &= w_{f_1}^T\varphi_{f_1}(x_1)\\
\hat{g}(x \mid w_g) &= w_{g_n}^T\varphi_{g_n}(x_n)\\
x_{n-1} &= w_{g_{n-1}}^T\varphi_{g_{n-1}}(x_{n-1})\\
&\vdots\\
x_1 &= w_{g_1}^T\varphi_{g_1}(x_1)
\end{aligned}$$

Sea la siguiente ley de aprendizaje:

$$
\begin{aligned}
&\dot{w}_{f_n}^T(t) = -\gamma_f PB\varphi_{f_n}(x)\, e_n(t)\\
&\dot{w}_{f_{n-1}}^T(t) = -\gamma_f PB\varphi_{f_{n-1}}(x)\, e_{n-1}(t)\\
&\vdots\\
&\dot{w}_{f_1}^T(t) = -\gamma_f PB\varphi_{f_1}(x)\, e_1(t)\\
&e_n(t) = e(\cdot)\\
&e_p(t) = 2\frac{\widehat{y}_q - w_{i,q}}{b_q} z_{q,k} \frac{\widehat{y}_p - c_{i,q,p}}{\left(\sigma_{q,p}^i\right)^2} e_q(t)\\
&\dot{w}_{g_n}^T(t) = -\gamma_g PB\varphi_{g_n}(x)\, e_n(t)\\
&\dot{w}_{g_{n-1}}^T(t) = -\gamma_g PB\varphi_{g_{n-1}}(x)\, e_{n-1}(t)\\
&\vdots\\
&\dot{w}_{g_1}^T(t) = -\gamma_g PB\varphi_{g_1}(x)\, e_1(t)\\
&e_p(t) = 2\frac{\widehat{y}_q - w_{i,q}}{b_q} z_{q,k} \frac{\widehat{y}_p - c_{i,q,p}}{\left(\sigma_{q,p}^i\right)^2} e_q(t)
\end{aligned}
\tag{4.19}
$$

Utilizando la ley (4.19), no se puede garantizar que los parámetros w_f y w_g, sean acotados. Si w_f diverge a infinito, entonces el sistema HFCMAC $\hat{f}(x \mid w_f)$, podría incrementarse y resultar en un control no acotado u_I. Al desarrollar un sistema estable (donde todas las variables estén acotadas), se puede modificar la ley de adaptación tal que los parámetros sean acotados. Sean los conjuntos compactos Ω_f y Ω_g para w_f y w_g se definen como:

$$
\begin{aligned}
\Omega_f &= \{w_f \in R_f \mid |w_f| \leq M_f\}\\
\Omega_g &= \{w_g \in R_g \mid 0 < \epsilon \leq |w_g| \leq M_g\}
\end{aligned}
\tag{4.20}
$$

Donde ϵ, w_f y w_g son constantes, se requiere que $|w_g|$ sea acotado por abajo por $0 < \epsilon$, de (4.8) se pude observar que $\hat{g} = w_g^T \varphi_g(x)$, es diferente de cero. Se modifican las leyes de (4.19) para garantizar que $w_f \in \Omega_f$ y $w_g \in \Omega_g$.

Si el vector de parámetros está en el interior del conjunto compacto ó en los límites, pero moviéndose hacia dentro del conjunto compacto, entonces se utiliza la ley (4.19); si el vector de parámetros está sobre el límite del conjunto compacto, pero moviéndose hacia afuera de él, entonces se emplea la proyección del gradiente del vector de parámetros sobre

el hiperplano de soporte. Las leyes de adaptación modificadas con proyección para el sistema neuro-difuso adaptable indirecto se explica a continuación:

Para w_f se define w_σ, $\sigma = 1, 2 \cdots n$, usando:

$$\dot{w}_\sigma(t) = \begin{cases} -\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t) & \text{si } |w_\sigma| < M_f \\ -\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t) \quad \text{si } PBw_\sigma^T\varphi_\sigma(x)\, e_\sigma(t) \geq 0 & \text{si } |w_\sigma| = M_f \\ T_\sigma\left(-\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t)\right) \quad \text{si } PBw_\sigma^T\varphi_\sigma(x)\, e_n(t) < 0 & \end{cases} \tag{4.21}$$

donde el operador de proyección $T\{\cdot\}$ se define como:

$$T_\sigma\left\{-\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t)\right\} = -\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t) + \gamma_\sigma PBe_\sigma(t)\, \frac{w_\sigma w_\sigma^T \varphi_\sigma(x)}{|w_\sigma|^2}$$

Para w_g

$$\dot{w}_\sigma(t) = \begin{cases} -\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t) & \text{si } PBw_\sigma^T\varphi_\sigma(x)\, e_\sigma(t) < 0 \\ 0 & \text{si } PBw_\sigma^T\varphi_\sigma(x)\, e_\sigma(t) \geq 0 \end{cases} \quad \text{si } w_\sigma = \epsilon$$

$$\dot{w}_\sigma(t) = \begin{cases} -\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t) & \text{si } |w_\sigma| < M_f \\ -\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t) \quad \text{si } PBw_\sigma^T\varphi_\sigma(x)\, e_\sigma(t) \geq 0 & \text{si } |w_\sigma| = M_f \\ T_\sigma\left(-\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t)\right) \quad \text{si } PBw_\sigma^T\varphi_\sigma(x)\, e_n(t) < 0 & \end{cases} \quad \text{en otro caso} \tag{4.22}$$

Donde ε es una constante positiva pequeña.

Teorema 4.1 *Sean los conjuntos compactos Ω_f y Ω_g definidos en (4.20). Si los valores iniciales de los parámetros satisfacen $w_f(0) \in \Omega_f$ y $w_g(0) \in \Omega_g$, entonces las leyes (4.21) y (4.22) garantizan que $w_f(t) \in \Omega_f$ y $w_g(t) \in \Omega_g$, para todo $t \geq 0$.*

Demostración. Para demostrar $|w_\sigma| \leq M_f$, se elije la siguiente función candidata de Lyapunov:

$$V_f = \frac{1}{2} w_\sigma^T w_\sigma \tag{4.23}$$

La derivada de la función candidata de Lyapunov es:

$$\dot{V}_f = w_\sigma^T \dot{w}_\sigma \tag{4.24}$$

Cuando $|w_\sigma| > M_f$, se tiene $|w_\sigma| < M_f$. La derivada de la función candidata de lyapunov es:

$$\dot{V}_f = -PBw_\sigma^T \varphi_\sigma(x)\, e_\sigma(t)$$

Cuando $|w_\sigma| = M_f$, si $PBw_\sigma^T \varphi_\sigma(x)\, e_\sigma(t) \geq 0$, entonces $\dot{V}_f \leq 0$. De aqui, se tiene $|w_f| \leq M_f$. Cuando $|w_\sigma| = M_f$, si $PBw_\sigma^T \varphi_\sigma(x)\, e_n(t) < 0$, entonces

$$\dot{V}_f = w_\sigma^T \left[-\gamma_\sigma PB\varphi_\sigma(x)\, e_\sigma(t) + \gamma_\sigma PBe_\sigma(t) \frac{w_\sigma w_\sigma^T \varphi_\sigma(x)}{|w_\sigma|^2} \right] = 0 \tag{4.25}$$

por lo que $|w_f| \leq M_f$. ■

Utilizando un análisis similar, se puede provar $w_g(t)$ que también es acotado para todo $t \geq 0$.

Si $|w_g| = \epsilon$, si $PBw_\sigma^T \varphi_\sigma(x)\, e_\sigma(t) < 0$ entonces:

$$\dot{V}_g = -PBw_\sigma^T \varphi_\sigma(x)\, e_\sigma(t) > 0 \tag{4.26}$$

Así que $|w_g|$ se incremente. si $PBw_\sigma^T \varphi_\sigma(x)\, e_\sigma(t) \geq 0$, w_g es fijo. entonces $|w_g| \geq \epsilon$.

Teorema 4.2 *Se considera un sistema dinámico del tipo 4.2 y dado un control adaptable HFCMAC como en 4.8, 4.9 con la ley adaptable 4.19, la estabilidad del sistema de control en lazo se garantiza, i.e., el error de trayectoria e es acotado y converge a:*

$$\limsup_{T\to\infty} \frac{1}{T} \int_0^T |e|_{Q_0}^2\, dt \leq \bar{\xi} \tag{4.27}$$

donde $Q_0 = (Q - \Pi)$, $\xi^T PB\Pi^{-1} PB\xi \leq \bar{\xi}$.

Demostración. Considere la siguiente función candidata de Lyapunov:

$$V = e^T Pe + \frac{1}{2\gamma_f} \tilde{w}_f^T \tilde{w}_f + \frac{1}{2\gamma_g} \tilde{w}_g^T \tilde{w}_g \tag{4.28}$$

Es obvio que V es una función definida positiva para cualquier t, $V(t) \geq 0$. De hecho, $V(t)$ representa una medida euclidiana variante en el tiempo de la distancia de la salida de la planta con respecto a la señal de referencia más la distancia de los parámetros del

controlador HFCMAC de sus valores óptimos. i.e. $e(t) = 0$, $w_f^T(t) = w_f^{*T}$, y $w_g^T(t) = w_g^{*T}$. Los cuáles γ_f, γ_g, son constantes positivas y P es una matriz definida positiva que satisface la ecuación de Lyapunov, con la ley de aprendizaje, donde $B = [0, 0 \cdots 0, 1]^T$, P es la solución de e conocida, tal que existe una matriz simétrica P ($P = P^T$) definida positiva de $n \times n$, la cuál satisface la ecuación de Lyapunov:

$$\Lambda^T P + P\Lambda = -Q, \quad Q = Q^T > 0 \tag{4.29}$$

Donde Q es una matriz de $n \times n$ definida positiva, y Λ está dado por (4.6). Con (4.29) la derivada en el tiempo de V a lo largo de la trayectoria del sistema en lazo cerrado puede ser derivado como:

$$\begin{aligned} \dot{V} &= -e^T Q e + e^T PB\xi + \frac{1}{\gamma_f}\tilde{w}_f^T\left[\dot{w}_f^T + \gamma_f^T e PB\varphi_{f_n}(x)\right] \\ &\quad + \frac{1}{\gamma_g}\tilde{w}_g^T\left[\dot{w}_g^T + \gamma_g e^T PB\varphi_{g_n}(x)\, u_I\right] \end{aligned}$$

Con (4.19) y

$$\dot{V} = -e^T Q e + e^T PB\xi \tag{4.30}$$

Donde $\xi = \epsilon_f + \epsilon_g u_I$. Utilizando la desigualdad matricial

$$X^T Y + \left(X^T Y\right)^T \leq X^T \Pi X + Y^T \Pi^{-1} Y \tag{4.31}$$

donde $X, Y, \Pi \in \Re^{n\times k}$ son cualquier matriz, Π es cualquier matriz definida positiva, se tiene:

$$e^T PB\xi \leq e^T \Pi e + \xi^T PB\Pi^{-1} PB\xi \tag{4.32}$$

donde $\Pi = \Pi^T > 0$, $\Pi < Q$. Porque u_I es acotado (w_f y w_g son acotados)

$$\xi^T PB\Pi^{-1} PB\xi \leq \bar{\xi}$$

El cuál puede ser representado como:

$$\dot{V} \leq -e^T (Q - \Pi) e + \bar{\xi} = -Q_0 \|e\|^2 + \beta_{\|\xi\|} \|\xi\| \tag{4.33}$$

donde $Q_0 = (Q - \Pi)$ y β son $\mathcal{K}_\infty$ funciones, V es una función de Lyapunov ISS. Utilizando el Teorema 4.2, la dinámica del error de identificación e es una entrada de estado estable. De aquí ξ está acotado por $\bar{\xi}$, la nueva entrada e es acotada.

Integrando (4.33) de 0 hasta T :

$$V_T - V_0 \leq -\int_0^T e^T Q_0 e dt + \bar{\xi} T$$

Por lo que:

$$\frac{1}{T}\int_0^T e^T Q_0 e dt \leq V_0 - V_T + \bar{\xi} T \leq \frac{V_0}{T} + \bar{\xi} \tag{4.34}$$

(4.27) es estable. ■

4.2.3. Simulaciones

El sistema barra-esfera se muestra en la figura 4.1, consiste de una barra horizontal sobre la cuál se desplaza una esfera, el objetivo del sistema es ubicar la esfera en una posición dada al modificar el ángulo θ. En este ejemplo se realizó un control de posición de la esfera sobre la barra, sin embargo con este sistema se puede atacar el problema de regulación y el de trayectoria.

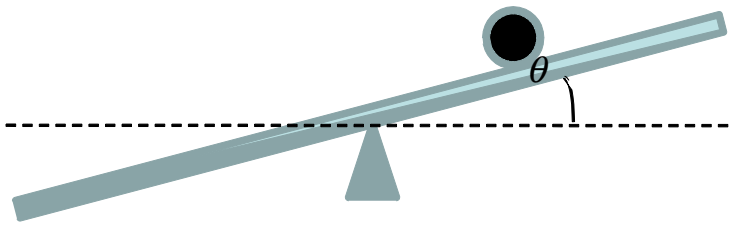

Figura 4.1: Sistema barra-esfera

El sistema barra-esfera tiene la siguiente representación matemática en espacio de estados:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \alpha\left(x_1 x_4^2 - \beta \sin x_3\right) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$
$$y = x_1$$

Donde $x = (x_1, x_2, x_3, x_4)^T = \left(r, \dot{r}, \theta, \dot{\theta}\right)^T$ es el vector de estados del sistema, r es la posición de la esfera, θ es el ángulo de la barra con respecto a la horizontal y $y = r$ es la salida del sistema. El control u es igual a la aceleración de θ. Los valores nominales del sistema barra-esfera son: $\alpha = 0{,}71$, $\beta = 9{,}8$, $k = 5/7$, $k_g = 75$, $r_s = 3{,}175 \times 10^{-2} m$, $L = 0{,}405 m$, $k_m = 0{,}0075 Nm/amp$, $k_e = 0{,}0075 volts - s/rad$, $R_a = 9\Omega$, $B_m = 1{,}6 \times 10^{-3} Nm - s/rad$, $Jm = 7{,}35 \times 10^{-3} Nm - s^2/rad$. Se utiliza un control indirecto HFCMAC.

Las simulaciones del sistema de control mediante la red HFCMAC se muestran en la figura 4.2, con este esquema de control se pueden compensar las incertidumbres tales como la fricción y la gravedad, además se puede ajustar la acción de control ya que presentan una ley adaptable. En la figura 4.2 está representado el control de posición de la esfera y en la figura 4.3 se presenta el error instantáneo. Los resultados muestran que el controlador adaptable HFCMAC puede controlar un sistema no lineal.

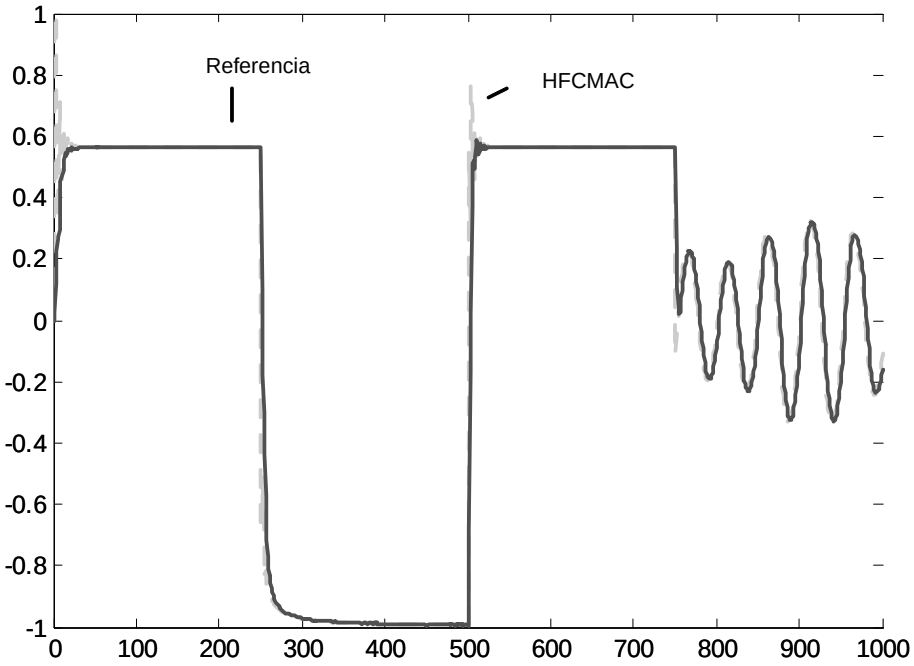

Figura 4.2: Respuesta del sistema de control.

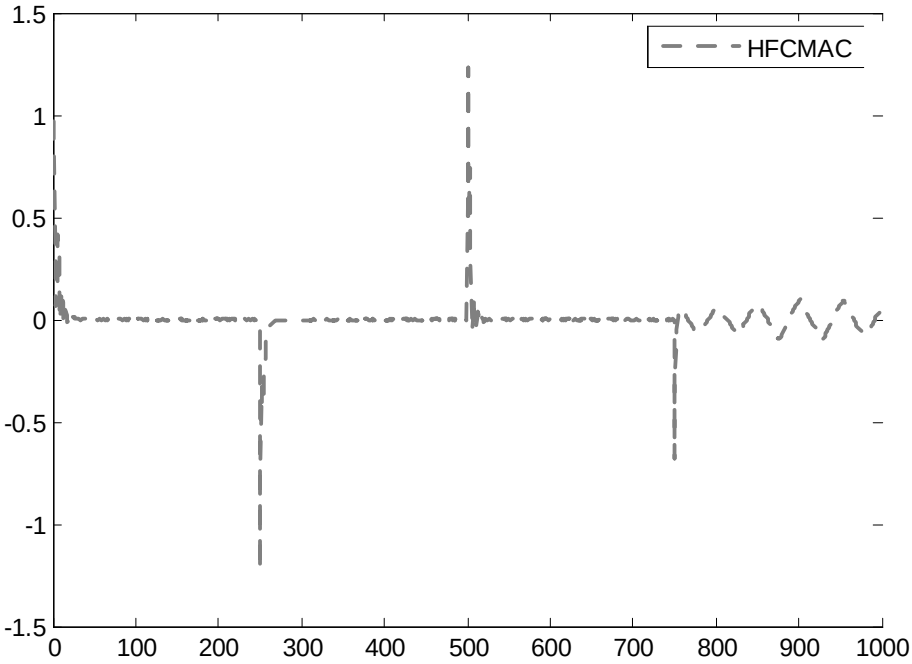

Figura 4.3: Error de control.

Capítulo 5

Conclusiones

5.1. Conclusiones

Se diseñaron diferentes esquemas de identificación para sistemas no lineales a partir de la red neuronal CMAC propuesta inicialmente por los trabajos de J. Albus en [1] y [2], se realiza una extensión combinando redes neuronales y sistemas difusos surgiendo así la red neuronal CMAC difusa (FCMAC).

La red FCMAC combina ambas técnicas: la capacidad de adaptación de la red neuronal y el conocimiento a priori de un experto en las reglas de un sistema difuso. Esta red presenta un ajuste muy rápido de sus pesos, ya que a diferencia de la red ANFIS que retropropaga el error a cada una de sus neuronas en las diferentes capas que lo componen para ajustar sus pesos, la red FCMAC solo ajusta los pesos de las neuronas activadas por las entradas. La red FCMAC presenta dos limitantes:

- Está limitado a sistemas estáticos ya que es una red hacia adelante.
- Para sistemas que presentan entradas $n - dimensionales$, $s_i \geq 3$, se vuelven difíciles de diseñar e implementar pues generan muchas reglas de inferencia.

Para resolver la primer limitante, en este trabajo se propone una nueva topologia de red recurrente FCMAC (RFCMAC). La primer topología RFCMAC presenta una retroal-

imentación local en la capa dos, la segunda topología tiene una retroalimentación global salida-entrada y una tercera presenta la combinación de ambas retroalimentaciones local más global. El algoritmo de actualización de los pesos está basado en el algoritmo de retropropagación del error (backpropagation). Se realiza un análisis vía segundo método de Lyapunov para demostrar estabilidad del sistema.

Para resolver la segunda limitante, se propone una nueva topología de red neuro-difusa CMAC con estructura jerárquica (HFCMAC), el cuál no presenta el problema del incremento en las reglas difusas al aumentar el número de entradas en la red ó al aumentar las funciones de pertenencia para cada entrada. Se diseñan las redes HFCMAC con entradas de baja dimensionalidad. Sin embargo con esta nueva topología únicamente se limita a sistemas estáticos, para poder realizar una identificación de sistemas dinámicos se propone una estructura jerárquica recurrente FCMAC (HFCMAC). Ambas topologías con estructura jerárquica presentan el algoritmo de entrenamiento de retro-propagación, se realiza el análisis de estabilidad de Lyapunov para asegurar la convergencia de los pesos.

La red CMAC con sus respectivas extenciones es capaz de controlar sistemas no lineales. Específicamente se diseñó un esquema de control adaptable indirecto HRFCMAC que realiza el control de un sistema barra-esfera, se realiza también una prueba de estabilidad del sistema.

Algunas de las aplicaciones de la red neuronal artificial CMAC incluye: Identificación de sistemas no lineales estáticos y dinámicos; Reconocimiento de patrones; Procesamiento digital de señales; Control de sistemas no lineales en tiempo real, entre otros.

Algunos tópicos de interés generados en este trabajo y que pueden ser estudiados más adelante son:

Realizar un análisis más profundo en las variables intermedias que se encuentran entre los niveles de la estructura jerárquica, ya que el diseño de la misma al carecer de un significado físico resulta difícil de implementar. Diseñar un control adaptable directo mediante redes HFCMAC para sistemas no lineales con múltiples entradas y múltiples salidas.

Bibliografía

[1] Albus, J. S. (1975 a), Data storage in the Cerebellar Model Articulatión Controller, Transactions of the ASME Journal of Dynamic Systems, Measurement and Control, Vol. 97, series G, No.3, pp. 228-233, September 1975.

[2] J. S. Albus, A New Approach to Manipulator Control: the Cerebellar Model Articulation Controller (CMAC), Transactions of the ASME Journal of Dynamic Systems, Measurement and Control, Vol. 97, series G, No.3, pp. 220-227, September 1975.

[3] A. M. Lyapunov, The General Problem of the Stability of Motion. Kharkov, Russia: Kharkov Math. Soc., 1892.

[4] Martin Brown, Chris Harris. Neurofuzzy adaptive modelling and control. Prentice Hall International. ISBN 0-13-134453-6. 1994.

[5] Commuri, S. and F. L. Lewis (1995a), Control of unknown nonlinear dynamical systems using CMAC neural networks: Structure, Stability and passivity. IEEE International symposium on Intelligent Control, San francisco, CA, pp. 123-129.

[6] Chih Min Lin and Ya Fu Peng, Adaptive CMAC based supervisory control for uncertain nonlinear systems. IEEE Transactions on Systems, Man and Cybernetics, Vol. 34, No. 2, April 2004.

[7] C.-T. Chiang and C.-S. Lin, CMAC with general basis functions, Neural Networks, vol. 9, no. 7, pp. 1199–1211, 1996.

[8] Ching-Hung Lee and Ching-Cheng Teng, Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks, IEEE Transactions on Fuzzy Systems, Vol. 8, No. 4, August 2000.

[9] J.-Y. Chen, P.-S. Tsai and C.-C. Wong, Adaptive design of a fuzzy cerebellar model arithmetic controller neural network, IEE Proc.-Control Theory Appl., Vol. 152, No. 2, Marzo 2005

[10] S. Commuri, F. L. Lewis. CMAC Neural Networks for Control of Nonlinear Dynamical Systems: Structure, Stability and Passivity. Automática, Vol. 33, No. 4, pp. 635-641, 1997.

[11] Francisco J. González Serrano, Aníbal R. Figueiras vidal and Antonio Artés Rodríguez. Generalizing CMAC Architecture and Training. IEEE Transactions on Neural Networks. Vol. 9. No. 6. pp. 1509-1514, November 1998.

[12] Ming-Ling Lee, Hung-Yuan Chung, Fang-Ming Yu. Modeling of hierarchical fuzzy systems. Fuzzy Sets and Systems 138 (2003) 343–361. Department of Electrical Engineering, National Central University, Taiwan.

[13] Jen Yang chen, An Adaptive FCMAC controller, Electronic Engineering Department. China Institute of Technology.

[14] Ken-Ichi Funahashi. On the approximate realization of continuos Mappings by Neural Networks, Neural Networks. pp. 183-192, Vol. 2, 1989.

[15] Kevin M. Passino. Intelligent Control: A Tutorial, IEEE Joint Int. Conference on Control Applications & Int. Symp. on Intelligent Control CCAISIC2001. México, DF., Sept. 5-7, 2001.

[16] Kumpati S. Narendra, Kannan Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks, IEEE Transactions on Neural Networks. Vol. 1. No. 1. pp. 4-27, March 1990.

[17] Kurkt Hornit. Multilayer Feedforward Networks are Universal Approximators. Neural Networks. pp. 359-366, Vol. 2, 1989.

[18] Kurkt Hornit. Approximation Capabilities of Multilayer Feedforward Networks. Neural Networks. pp. 251-257, Vol. 4, 1991.

[19] L. Zadeh, Fuzzy sets, inform. Contr., Vol. 8, No. 3, pp. 338-353, 1965.

[20] Young H. Kim, Frank L. Lewis. Optimal Design of CMAC Neural Network Controller for Robot Manipulators. IEEE Transactions on Systems, Man and Cybernetics. Vol. 30. No. 1. February 2000.

[21] Filson H. Glanz, W. Thomas Miller, L. gordon Kraft. An overview of the CMAC neural network. Robotics Laboratory, Electrical Engineering Department.

[22] Qiang Gan, Eric M. Rosales, A Comparative Study on CMAC and ANFIS for Nonlinear System Modelling, Technical Report, Department of Computer Science, University of Essex, 2002.

[23] G.V.S. Raju, Jun Zhou, and R. A. Kisner, Hierarchical fuzzy Control, International Joint Control., Vol. 54, No. 5, pp. 1201 -1216 , 1991.

[24] G.V.S. Raju, Jun Zhou, Adaptive Hierarchical Fuzzy Controller, IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 4, July-August 1993.

[25] Rong-Jong Wai, Chih-Min Lin, and Ya-Fu Peng, Adaptive Hybrid Control for Linear Piezoelectric Ceramic Motor Drive Using Diagonal Recurrent CMAC Network, IEEE Transactions on Neural Networks, vol. 15, No. 6, November 2004.

[26] P. S. Sastry, G. Santharam, and K. P. Unnikrishnan, Memory neural networks for identification and control of dynamic systems, IEEE Trans. Neural Networks, vol. 5, 306-319, 1994.

[27] Li-Xin Wang. Analysis and Design of Hierarchical Fuzzy Systems. IEEE Transactions on Fuzzy Systems, vol. 7, No. 5, October 1999.

[28] C. Wei and L.X. Wang, A Note on Universal Approximation by Hierarchical Fuzzy Systems, *Information Sciences*, Vol. 123, 241-248, 2000.

[29] Xiao-Jun Zeng and John A. Keane, Approximation Capabilities of Hierarchical Fuzzy Systems, IEEE Transactions on Fuzzy Systems, Vol. 13, No. 5, October 2005.

[30] Zhong Ping Jiang, Yuan Wang, Input to state stability for discrete time nonlinear systems, Automática 37, 857-869, 2001.

[31] Russell L. Smith. Intelligent Motion Control with an Artificial Cerebellum. Ph. D. dissertation, The Department of Electrical and Electronic Engineering. University of Auckland, New Zealand. July 1998.

[32] Modular on line function approximation for scaling up reinforcement learning, Chen Khong Tham, University of Cambrige, England. October 1994, Thesis doctoral.

[33] Mehrdad Hojati, Saeed Gazor, Hybrid Adaptive Fuzzy Identification and Control of Nonlinear Systems, IEEE, Transactions on fuzzy systems, Vol. 10, No. 2 April 2002.

Capítulo 6

Apéndices

6.1. A. Preliminares Matemáticos

6.1.1. Norma Euclideana

La norma euclideana $\|x\|$ de un vector $x \in \mathbb{R}^n$ se define como:

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x^T x}$$

donde sólo se considera la parte positiva de la raíz cuadrada. Es inmediato comprobar que la norma euclideana satisface las siguientes propiedades:

- $\|x\| = 0$, si y sólo si $x = 0 \in \mathbb{R}^n$.
- $\|x\| > 0$, para todo $x \in \mathbb{R}^n$ con $x \neq 0 \in \mathbb{R}^n$.
- $\|\alpha x\| = |\alpha| \, \|x\|$, para todo $\alpha \in \mathbb{R}$ y $x \in \mathbb{R}^n$.
- $\|x\| - \|y\| \leq \|x + y\| \leq \|x\| + \|y\|$, para todo $x, y \in \mathbb{R}^n$.
- $\left|x^T y\right| \leq \|x\| \, \|y\|$, para todo $x, y \in \mathbb{R}^n$ (desigualdad de Schwarz).

6.1.2. Matrices

Se denotará por $\mathbb{R}^{n\times m}$ al conjunto de matrices A de dimensión $n\times m$ formada por arreglos de números reales ordenados por n renglones y m columnas:

$$A = \{a_{ij}\} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Un vector $x \in \mathbb{R}^n$ puede ser interpretado como una matriz particular perteneciente a $\mathbb{R}^{n\times 1} = \mathbb{R}^n$. La matriz traspuesta $A^T = \{a_{ji}\} \in \mathbb{R}^{m\times n}$ se obtiene intercambiando los renglones y columnas de $A = \{a_{ij}\} \in \mathbb{R}^{n\times m}$.

Producto de matrices

Considérense las matrices $A \in \mathbb{R}^{m\times p}$ y $B \in \mathbb{R}^{p\times n}$. El producto de las matrices A y B denotado por $C = AB \in \mathbb{R}^{m\times n}$ se define como:

$$\begin{aligned} C &= \{c_{ij}\} = AB \\ &= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mp} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pn} \end{bmatrix} \end{aligned}$$

Puede verificarse que el producto de matrices satisface las siguientes propiedades:

- $(AB)^T = B^T A^T$, para toda $A \in \mathbb{R}^{m\times p}$ y $B \in \mathbb{R}^{p\times n}$.
- $AB \neq BA$, en general.
- $A(B + C) = AB + AC$, para toda $A \in \mathbb{R}^{m\times p}$, $B \in \mathbb{R}^{p\times n}$ y $C \in \mathbb{R}^{p\times n}$.

- $ABC = A(BC) = (AB)C$ para toda $A \in \mathbb{R}^{m\times p}$, $B \in \mathbb{R}^{p\times n}$ y $C \in \mathbb{R}^{n\times r}$.

De acuerdo con la definición del producto de matrices, la expresión $x^T Ay$ donde $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times m}$ y $y \in \mathbb{R}^m$ está dado por:

$$\begin{aligned} x^T Ay &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}x_i y_j \end{aligned}$$

Matrices especiales

Una matriz A es cuadrada si $n = m$. Una matriz cuadrada $A \in \mathbb{R}^{n\times n}$ es simétrica si es igual a su transpuesta $A = A^T$, es antisimétrica si $A = -A^T$. Una propiedad de las matrices antisimétricas es:

$$x^T Ax = 0,\ \forall x \in \mathbb{R}^n$$

Una matriz cuadrada $A = \{a_{ij}\} \in \mathbb{R}^{n\times n}$ es diagonal si $a_{ij} = 0$ para todo $i \neq j$. Se denota por:

$$diag\{a_{11}, a_{22}, ..., a_{nn}\} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

Una matriz cuadrada $A \in \mathbb{R}^{n\times n}$ es singular si su determinante es nulo, i.e., si $det(A) = 0$, en caso contrario es no singular. Una característica de una matriz singular es que ésta no tiene inversa.

Una matriz cuadrada $A \in \mathbb{R}^{n\times n}$, sin ser necesariamente simétrica, es definida positiva si:

$$x^T Ax > 0,\ \forall x \in \mathbb{R}^n,\ \text{con } x \neq 0 \in \mathbb{R}^n$$

Cualquier matriz simétrica y definida positiva $A = A^T > 0$ es no singular, por lo tanto su inversa A^{-1} existe. Una matriz cuadrada $A \in \mathbb{R}^{n\times n}$, es semidefinida positiva si:

$$x^T Ax \geq 0,\ \forall x \in \mathbb{R}^n$$

Una matriz cuadrada $A \in \mathbb{R}^{n\times n}$, es definida negativa si $-A$ es definida positiva, mientras que es semidefinida negativa si $-A$ es semidefinida positiva.

6.1.3. Valores propios

Para cada matriz cuadrada $A \in \mathbb{R}^{n\times n}$ existen n valores propios (números complejos en general), denotados por $\lambda_1\{A\}, \lambda_2\{A\}, ..., \lambda_n\{A\}$. Los valores propios de la matriz $A \in \mathbb{R}^{n\times n}$ satisfacen:

$$\det[\lambda_i\{A\} I - A] = 0,\ i = 1, 2, ..., n$$

donde $I \in \mathbb{R}^{n\times n}$ es la matriz identidad de dimensión n. Para el caso de una matriz simétrica $A = A^T \in \mathbb{R}^{n\times n}$, sus valores propios son tales que:

- $\lambda_1\{A\}, \lambda_2\{A\}, ..., \lambda_n\{A\}$ son números reales.
- $\lambda_i\{A\}$ denota el valor propio de la matriz A. $\lambda_{\text{máx}}\{A\}$ denota el valor propio más grande de la matriz A si todos los valores propios son reales. $\lambda_{\text{mín}}\{A\}$ denota el valor propio más pequeño de la matriz A si todos los valores propios son reales. El teorema de Rayleigh-Ritz establece que para todo $x \in \mathbb{R}^n$ se tiene:

$$\lambda_{\text{mín}}\{A\}\|x\|^2 \leq x^T Ax \leq \lambda_{\text{máx}}\{A\}\|x\|^2$$

Una matriz cuadrada $A \in \mathbb{R}^{n\times n}$, es definida positiva si y sólo si los valores propios de $A + A^T$ son positivos, i.e., $\lambda_i\{A + A^T\} > 0$ para $i = 1, 2, ..., n$. Más aún, una matriz simétrica $A = A^T \in \mathbb{R}^{n\times n}$ es definida positiva si y sólo si $\lambda_i\{A\} > 0$ para $i = 1, 2, ..., n$.

6.1.4. Norma espectral

La norma espectral $\|A\|$ de una matriz $A \in \mathbb{R}^{n\times n}$, se define como:

$$\|A\| = \sqrt{\lambda_{\text{máx}}\left\{A^T A\right\}}$$

donde $\lambda_{\text{máx}}\left\{A^T A\right\}$ denota el valor propio máximo de la matriz simétrica $A^T A \in \mathbb{R}^{m\times m}$. En el caso particular de matrices simétricas $A = A^T \in \mathbb{R}^{n\times n}$, se tiene que:

- $\|A\| = \text{máx}_i \left|\lambda_i \left\{A\right\}\right|$.
- $\|A^{-1}\| = \frac{1}{\text{mín}_i|\lambda_i\{A\}|}$.

En las expresiones anteriores el valor absoluto resulta rebundane si A es simétrica y definida positiva $A = A^T > 0$. La norma espectral satisface las siguientes propiedades:

- $\|A\| = 0$, si y sólo si $A = 0 \in \mathbb{R}^{n\times m}$.
- $\|A\| > 0$, para todo $A \in \mathbb{R}^{n\times m}$ con $A \neq 0 \in \mathbb{R}^{n\times m}$.
- $\|A + B\| \leq \|A\| + \|B\|$, para todo $A, B \in \mathbb{R}^{n\times m}$.
- $\|\alpha A\| = |\alpha| \|A\|$, para todo $\alpha \in \mathbb{R}$ y $A \in \mathbb{R}^{n\times m}$.
- $\left\|A^T B\right\| \leq \|A\| \|B\|$, para todo $A, B \in \mathbb{R}^{n\times m}$.

Un resultado particular es la matriz $A \in \mathbb{R}^{n\times m}$ y el vector $x \in \mathbb{R}^m$. La norma euclideana del vector Ax satisface:

$$\|Ax\| \leq \|A\| \|x\|$$

donde $\|A\|$ denota la norma espectral de la matriz A, mientras que $\|x\|$ denota la norma euclideana del vector x. Siendo $y \in \mathbb{R}^n$, el valor absoluto de $y^T Ax$ satisface satisface:

$$\left\|y^T Ax\right\| \leq \|A\| \|y\| \|x\|.$$

6.1.5. Supremo e ínfimo

Una de las propiedades fundamentales de los números reales es la existencia del supremo y del ínfimo para subconjuntos acotados.

Definición 6.1 *Sea $A \subseteq \mathbb{R}$. A se dice acotado superiormente si existe $M \in \mathbb{R}$ tal que $\forall r \in A$ se tiene $r \leq M$. Sea A un subconjunto no vacío de los reales que es acotado superiormente de define el supremo de A ($supA$) como el mínimo número real M tal que $r \leq M$ $\forall x \in A$,en caso de existir tal M. El conjunto A se dice acotado inferiormente si existe $N \in \mathbb{R}$ tal que $\forall r \in A$ se tiene $N \leq r$. Si A es un subconjunto no vacío de los números reales, el cuál es acotado inferiormente, se define el ínfimo de A ($\inf A$) como el máximo número real N tal que $N \leq r$ $\forall x \in A$, en caso de existir tal N.*

Axioma 6.1 *Sea $A \neq 0$, $A \subseteq \mathbb{R}$. Si A es acotado superiormente, entonces existe sup $A \in \mathbb{R}$. Equivalentemente, si $A \neq 0$, $A \subseteq \mathbb{R}$. Si A es acotado inferiormente, entonces existe inf $A \in \mathbb{R}$.*

6.2. B Algoritmo de aprendizaje

6.2.1. Mínimos cuadrados

El algoritmo de mínimos cuadrados ($least - mean - square$ LMS) está basado en el valor instantáneo de la función de costo definido por:

$$\xi(w) = \frac{1}{2} e^2(n)$$

donde $e(n)$ es la señal del error medida en el tiempo n. Derivando $\xi(w)$ con respecto al vector de pesos w se obtiene:

$$\frac{\partial \xi(w)}{\partial w} = e(n) \frac{\partial e(n)}{\partial w}$$

El algoritmo LMS opera con una neurona lineal, por lo que se puede expresar la señal de error como:

$$e(n) = d(n) - s^T(n) w(n)$$

por lo tanto

$$\frac{\partial e\left(n\right)}{\partial w\left(n\right)} = -s\left(n\right)$$

y

$$\begin{aligned} \frac{\partial \xi\left(w\right)}{\partial w} &= -e\left(n\right) s^{T}\left(n\right) \\ &= -s\left(n\right) e\left(n\right) \end{aligned}$$

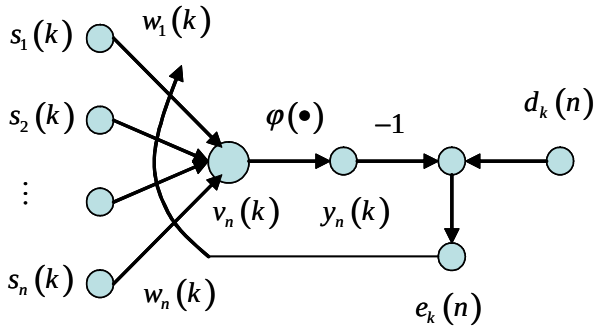

Figura 6.1: Propagación de la señal en una red neuronal

Utilizando éste último resultado como un estimado del vector gradiente, se puede escribir:

$$\hat{g}\left(n\right) = -s\left(n\right) e\left(n\right) \tag{6.1}$$

Por último, utilizando la ecuación 6.1 del vector gradiente para el algoritmo del paso descendente descrito por $w\left(n+1\right) = w\left(n\right) - \eta g\left(n\right)$, se puede formular el algoritmo LMS como:

$$\hat{w}\left(n+1\right) = \hat{w}\left(n\right) + \eta s\left(n\right) e\left(n\right) \tag{6.2}$$

donde η es la tasa de aprendizaje. El lazo de retroalimentación en torno al vector de peso $\hat{w}\left(n\right)$, se emplea $\hat{w}\left(n\right)$ en lugar de $w\left(n\right)$ para enfatizar el hecho de que el algoritmo LMS produce un estimado del vector de pesos. La presencia de la retroalimentación tiene un profundo impacto sobre el comportamiento de convergencia del algoritmo LMS. Del algoritmo

LMS 6.2 se puede observar que depende del término η, taza de aprendizaje y del vector de entrada $s(n)$, de aqui se dedude que la convergencia del algoritmo LMS depende del vector de entrada $s(n)$ y del valor asignado al parámetro de aprendizaje η. Un análisis estadistico del algoritmo LMS la cuál está basada en la llamada teoría de la independencia (Widro et al. 1976), la cuál asume que la taza de aprendizaje η es suficientemente pequeña tal que LMS converge al valor promedio cuadrado, la taza de aprendizaje η satisface la condición:

$$0 < \eta < \frac{2}{\lambda_{máx}}$$

Donde $\lambda_{máx}$ es el eigenvalor más grande de una matriz de correlación R_x, sin embargo en la mayoría de las aplicaciones $\lambda_{máx}$ no está disponible. Por lo que la traza de la matriz R_x se puede tomar como un estimado de $\lambda_{máx}$, por lo que la condición anterior se puede reescribir como:

$$0 < \eta < \frac{2}{T_r[R_x]}$$

Por definición la traza de una matriz cuadrada es igual a la suma de los elementos de su diagonal. En términos matemáticos el algoritmo LMS es óptimo de acuerdo a los criterios de H_∞. Una de las principales limitaciones del algoritmo LMS es que presenta una taza de convergencia lenta y es muy sensible a las variaciones de los eigenvalores de la entrada. El algoritmo LMS requiere aproximadamente de un número de iteraciones 10 veces la dimensión del espacio de entrada para alcanzar la condición en estado estacionario. La taza de convergencia lenta llega a ser un problema cuando la dimensionalidad del espacio de entrada es grande. Hay que hacer notar que el algoritmo LMS está diseñado para una neurona lineal. Las dificultades encontradas en el algoritmo LMS se debe al hecho de que el parámetro de aprendizaje se mantiene constante $\eta(n) = \eta_0 \, \forall n$, en cambio en una aproximación estocástica la taza de aprendizaje η es variante en el tiempo, descrita comúnmente por:

$$\eta(n) = \frac{c}{n}$$

donde c es una constante, lo cuál es una elección suficiente para garantizar la convergencia en la aproximación estocástica, como una alternativa a las ecuaciones anteriores se puede

utilizar la expresión:

$$\eta(n) = \frac{\eta_0}{1 + (n/\tau)}$$

Donde η_0 y τ son constantes seleccionadas por el usuario. Con estas modificaciones el algoritmo opera como un aproximador estocástico tradicional y la convergencia de los pesos llegan a ser óptimos.

6.2.2. Retropropagación

La señal de error en la salida de la neurona j en la iteración k está definido por:

$$e_j(k) = d_j(k) - y_j(k)$$

Se define el valor instantáneo del error para la neurona j como $\frac{1}{2}e_j^2(k)$, la suma de los errores instantáneos al cuadrado se formula como:

$$\varepsilon(n) = \frac{1}{2}\sum_{j \in C}^{l} e_j^2(k)$$

donde C es el conjunto de neuronas de salida, $C = \{1, , 2, ..., l\}$. El error promedio (e_{av}) se obtiene de promediar los errores instantáneos correspondientes a los N pares de entrenamiento.

$$\varepsilon_{av}(n) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon(k)$$

El objetivo es minimizar ε_{av} con respecto a los pesos. Se necesita calcular $\Delta w_{ji}(k)$.

$$\frac{\partial \varepsilon(k)}{\partial w_{ji}(k)}$$

Por la regla de la cadena se tiene:

$$\frac{\partial \varepsilon(k)}{\partial w_{ji}(k)} = \frac{\partial \varepsilon(k)}{\partial e_j(k)} \frac{\partial e_j(k)}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial v_j(k)} \frac{\partial v_j(k)}{\partial w_{ji}(k)} \tag{6.3}$$

por lo que:

$$v_j(k) = \sum_{i=0}^{p} w_{ji}(k)\, y_j(k)$$

$$y_j(k) = \varphi(v_j(k))$$

Se definen los componentes de 6.3 como sigue:

$$\begin{aligned} \frac{\partial \varepsilon(k)}{\partial e_j(k)} &= e_j(n) \\ \frac{\partial e_j(k)}{\partial y_j(k)} &= -1 \\ \frac{\partial y_j(k)}{\partial v_j(k)} &= \varphi_j(v_j(k)) \\ \frac{\partial v_j(k)}{\partial w_{ji}(k)} &= y_j(k) \end{aligned} \tag{6.4}$$

Sustituyendo 6.4 en 6.3 se tiene:

$$\frac{\partial \varepsilon(k)}{\partial w_{ji}(k)} = -e_j(k)\,\varphi_j(v_j(k))\,y_j(k)$$

Perturbaciones e incertidumbres

Las consideraciones sobre sensibilidad son importantes en el diseño de sistemas de control, ya que todos los elementos físicos tienen la propiedad de cambiar con el ambiente y con el tiempo, no se pueden considerar a los parámetros de un sistema de control completamente estacionarios durante toda su vida de operación, todos los sistemas físicos están sujetos a señales externas ó ruido durante su funcionamiento. Una perturbación es una señal que tiende a afectar negativamente el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se denomina interna, en tanto que una perturbación externa se produce fuera del sistema es una entrada. En general, un buen sistema de control debe ser insensible a la variación de los parámetros pero sensible a los comandos ó datos de entrada. El efecto de la realimentación sobre el ruido y la perturbación depende en mayor medida en que parte del sistema se presentan las señales externas, en la mayoria de las ocasiones la realimentación puede reducir los efectos del ruido y las perturbaciones en el desempeño del sistema.

6.2.3. Consideraciones básicas del algoritmo de aprendizaje

en esta sección se analizan algunas mejoras del algoritmo de retropropagación. Si se selecciona un parámetro de aprendizaje η pequeño, entonces el ajuste a los pesos sinápticosque se da de iteración a iteración será pequeño, por lo tanto el aprendizaje es lento. De lo contario si η se selecciona grande, entonces el aprendizaje será rápido, pero existe la posibilidad de que la red se vuelva inestable, se pueden presentar oscilaciones. Una forma sencilla de incrementar el parámetro de la velocidad de apendizaje de modo que se evite la posibilidad de inestabilidad consiste en modificar la regla delta, incluyendo un término de momento.

$$\Delta\omega_{ji}(k) = \alpha\Delta\omega_{ji}(k-1) + \eta\delta_j(k)\, y_i(k)$$

Basándose en esta ecuación, se pueden hacer los siguientes comentarios:

- Para que la serie de tiempo converga es necesario que la constante de momento se restringa a $0 \leq \alpha < 1$ cuando $\alpha = 0$ el algoritmo de retropropagación opera sin momento.
- Es posible que el parámetro de velocidad de aprendizaje η sea dependiente de la conexión y de la iteración $\eta_{ji}(k)$.
- Para el caso de que se tengan algunos pesos fijos se puede tomar $\eta_{ji}(k) = 0$.

El algoritmo de retropropagación se ha establecido como uno de los algoritmos más populares para el entrenamiento supervisado de redes neuronales tipo perceptrón multicapa. Básicamente es una técnica de gradiente y no de optimización, tiene dos propiedades distintivas:

- Es fácil de calcular localmente.
- Logra un gradiente descendiente estocástico.
- Su complejidad computacional es lineal.

Existen algunas heurísticas que proveen una guía adecuada para acelerar la convergencia del algoritmo de retropropagación por medio de la tasa de aprendizaje:

- Todos los parámetros ajustables de la red deben tener su propio parámetro de aprendizaje.
- Todo parámetro de aprendizaje debe variar de una iteración a otra.
- Cuando la derivada de la función de costo con respecto a los pesos sinápticos tiene el mismo signo algebraico para iteraciones consecutivas del algortimo, el parámetro de aprendizaje se debe incrementar.
- Cuando el signo algebraico de la derivada de la función de costo con respecto a los pesos sinápticos tiene diferente signo para iteraciones consecutivas del algoritmo, el parámetro de aprendizaje se debe de reducir.

Considerando estas heurísticas, el algoritmo de retropropagación ya no es más un algoritmo de gradiente descendente, ahora los ajustes se basan en: derivadas parciales de la superficie del error con respecto a los pesos; Estimados de la curvatura de la superficie del error en el punto de operación.

6.3. C. Esquemas de identificación

En el estudio de los sistemas una de las primeras necesidades es la de obtener modelos. Un modelo sintetiza el análisis de las principales caracteristicas del sistema y de su precisión depende la profundidad con la que se realiza. Para esto se deben detectar y determinar cuáles son las variables del sistema más relevantes y el tipo de interacciones causa-efecto que entre ellas se producen.

Los modelos de identificación son estructuras en las cuáles se ajustan sus parámetros con el fin de obtener una respuesta muy similar al del sistema en estudio, dentro de cierta zona de operación. En el desarrollo del presente capítulo se busca interpretar la identificación como la

evolución del concepto de aproximación de funciones, y se describe como las redes neuronales han proporcionado buenos resultados en este tipo de tareas, un comentario importante es que la utilización de las redes neuronales estáticas presenta las desventaja de crear modelos secuenciales, entrenados en un esquema serie-paralelo, por lo que hace que el proceso de ajuste de los pesos sea muy lento.

6.3.1. Aproximación de funciones

Se puede considerar el problema de aproximación de funciones como antecedente al de identificación de sistemas. El problema de aproximación de funciones se divide en dos tipos:

- En el primero, se supone que se conoce la expresión explícita de la función, y lo que se busca es una estructura paramétrica que aproxime la función con algún grado de precisión deseado.
- En el segundo caso, el problema consiste en obtener una función que se ajuste a la relación entrada-salida existente en un conjunto finito de datos.

En el primer caso se busca representar cierta clase de funciones como una suma de potencias a través de manipular series geométricas, como lo son las series de Mclaurin y de Taylor. Este enfoque permite obtener versiones simplificadas de funciones, con el único fin de ayudar al análisis de los problemas y también para simplificar los cálculos. En el segundo caso está enfocado de manera experimental debido a que no se cuenta con una expresión explícita de la función de la cuál se desea obtener una aproximación.

Se define el problema de aproximación de funciones de la siguiente manera [15]:

Definición 6.2 *Sea* $F(s,\theta)$ *una función de aproximación no lineal ajustable. El vector* $s = [s_1, s_2, ..., s_n]^T$ *es la entrada del aproximador y el vector* $\theta = [\theta_1, \theta_2, ..., \theta_p]^T$ *define sus parámetros, donde* F *es la función de aproximación. Sea* $G(s,z)$ *una función donde* $s = [s_1, s_2, ..., s_n]^T$ *es la entrada y* $z = [z_1, z_2, ..., z_n]^T$ *es una entrada auxiliar para* G*. El problema de la aproximación de funciones se divide en dos etapas, en la primera parte se*

debe proponer una estructura para la función F, y en la segunda parte se debe definir la forma para ajustar el vector de parámetros θ.

Para el caso de que no se conozca la función G de manera explícita, se define el problema de aproximación de funciones de la siguiente manera:

Definición 6.3 *Supóngase que en el $i-$ésimo experimento se tiene el vector de entrada:*

$$s(i) = [s_1(i), s_2(i), ..., s_n(i)]^T$$

con el vector auxiliar:

$$z(i) = [z_1(i), z_2(i), ..., z_n(i)]^T$$

donde la salida está dada por:

$$y(i) = G(s(i), z(i))$$

debido a que no se conoce de manera explícita a G se puede conocer experimentando y recolectando un conjunto de datos de entrada y salida. En la práctica se restringen estos experimentos a un subconjunto S del espacio de entradas, $s(i) \in S \subset R^n$ y de forma similar para la entrada auxiliar $z(i) \in Z \subset R^n$.

El par $(s(i), y(i))$ es un dato de entrenamiento, al conjunto de datos de entrenamiento de la forma:

$$\varsigma = [s(i), y(i), ..., s(M), y(M)]$$

en donde M denota el número de pares de entrenamiento contenidos en ς. El problema de aproximación de funciones consiste en hallar la forma de calcular el valor del vector de parámetros θ en $F(s, \theta)$ de tal forma que:

$$G(s, z) = F(s, \theta) + e(s, z)$$

en donde $e(s, z)$ es muy pequeño.

Un sistema dinámico puede ser descrito por dos tipos de modelos: modelos de entrada-salida y modelos en el espacio de estados. A continuación se describirá cada uno de estos modelos.

6.3.2. Modelo de entrada-salida

El modelo de entrada-salida describe a un sistema basado únicamente en los datos de entrada y salida que presenta. Para sistemas discretos, el modelo de entrada-salida puede ser de tipo $NARMAX$ ó de tipo paramétrico Hammerstein [16]. En el modelo de entrada-salida se asume que la salida del sistema puede ser calculada por sus entradas pasadas y por sus salidas actuales. Para un sistema deterministico, invariante en el tiempo, con una entrada - una salida, se tiene:

$$y_p(k) = f\left(\begin{array}{c} y_p(k-1), y_p(k-2), ..., y_p(k-n), \\ u(k-1), u(k-2), ..., u(k-m) \end{array}\right) \tag{6.5}$$

donde $(u(k), y_p(k))$ son los pares de entrada-salida del sistema discreto en k, n y m son enteros positivos y denotan el número de entradas y salidas pasadas, n es el orden del sistema. En la práctica, generalmente m es más pequeño ó igual a n. f es una función no lineal. Si 6.5 es un sistema lineal, f es una función lineal y puede reescribirse como:

$$y_p(k) = f\left(\begin{array}{c} a_1 y_p(k-1), a_2 y_p(k-2), ..., a_n y_p(k-n), \\ b_1 u(k-1), b_2 u(k-2), ..., b_m u(k-m) \end{array}\right) \tag{6.6}$$

donde a_i, $i = 1, 2, ..., n$ y b_j, $j = 1, 2, ..., m$ son constantes con valores reales.

La ecuación 6.5 puede ser representado por el diagrama a bloques de la figura 6.2. Se puede observar en la figura que cuando los datos de entrada-salida son usados, el sistema dinámico está definido por la función $f(\cdot)$ y los enteros m y n. Si m y n están dados, la tarea consiste en encontrar la función $f(\cdot)$, para sistemas invariantes en el tiempo $f(\cdot)$ será constante. De acuerdo a la configuración de identificación de sistemas, se pueden presentar dos estructuras de identificación: estructura paralela y estructura serie-paralela.

En la estructura paralela, la red neuronal y el sistema reciben las mismas entradas externas, las salidas del sistema no son usadas como entradas a la red. El sistema y la red neuronal son dos procesos independientes los cuáles presentan las mismas entradas externas y la salida de cada una de ellas no interfiere una con la otra, es decir, presentan salidas independientes, como se observa en la figura 6.3.

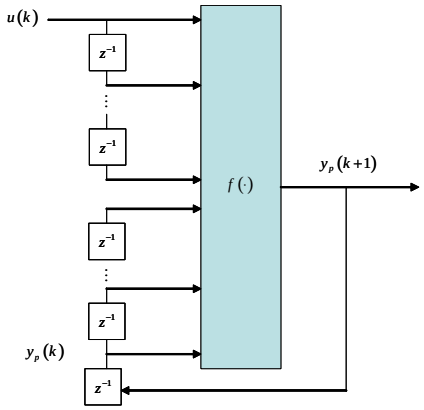

Figura 6.2: Modelo entrada-salida.

En la estructura serie-paralela la red y el sistema reciben las mismas entradas externas, pero la salida del sistema forma parte de las entradas a la red, el sistema y la red no son procesos independientes, el comportamiento dinámico de la red es afectado por las salidas del sistema, esto se puede observar en la figura 6.4.

Cuando los modelos paralelo y serie-paralelo son utilizados para la identificación, se asume que los sistemas presentan entradas y salidas acotadas, sin embargo la estructura paralela no garantiza que el proceso de aprendizaje de los pesos converga o que el error entre la salida del sistema y de la red tiendan a cero.

6.4. D Estabilidad

Se considera una ecuación diferencial ordinaria nolineal (ODE) en tiempo continúo:

$$\dot{x}(t) = f(x(t)) \tag{6.7}$$

con el estado $x \in \mathbb{R}^n$.

Para el estado inicial $x \in \mathbb{R}^n$ en el tiempo inicial $t = 0$, se denota la solucion de (6.7) por $\varphi(t, x)$,i.e:

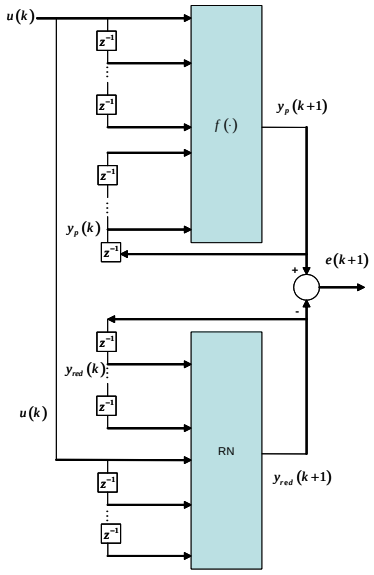

Figura 6.3: Identificación con estructura paralela.

- $x(t) = \varphi(t, x)$ es la solucion de ODE (6.7).
- $\varphi(0, x) = x$

Se asume la unicidad de $\varphi(t, x)$ sobre un intervalo y que 0 es un estado de equilibrio para ODE (6.7), i.e. $f(0) = 0$.

La ODE es llamada **estable asintóticamente globalmente** (GAS) si para todos los valores iniciales $x \in \mathbb{R}^n$ se tiene:

Estabilidad: para todo $\epsilon > 0$ existe $\delta > 0$ con

$$\|x\| < \delta \Rightarrow \|\varphi(t, x)\| < \epsilon$$

para todo $t \geq 0$.

Globalmente atractivo: para todo $\epsilon > 0$ y $r > 0$ existe $T > 0$ con

$$\|x\| < r \Rightarrow \|\varphi(t, x)\| < \epsilon$$

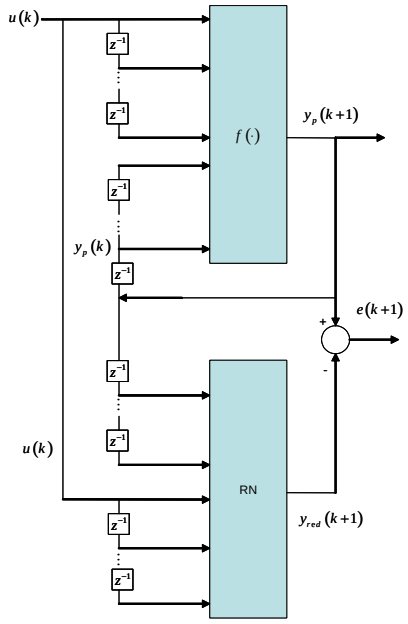

Figura 6.4: Identificación serie-paralela.

para todo $t \geq T$.

Más convenientemente se pude definir la estabilidad asintótica globalmente mediante la comparación de funciones, para esto es necesario que definamos algunas clases de funciones que nos serán de gran utilidad.

Definición 6.4 *Función clase $\mathcal{K}$. Si una función continua $\gamma(\cdot) : [0, a) \rightarrow [0, \infty)$ es estrictamente creciente con $\gamma(0) = 0$. $\gamma(\cdot)$ es llamada una función clase $\mathcal{K}$.*

Definición 6.5 *Si $\gamma(s)$ es una función clase $\mathcal{K}$, y además no está acotado, i.e. $\lim_{s\rightarrow\infty} \gamma(s) = \infty$ se dice que $\gamma(s)$ pertenece a la clase $\mathcal{K}_\infty$.*

Definición 6.6 *Función clase $\mathcal{KL}$. Una función continúa $\beta(s, t) : [0, a) \times [0, \infty) \rightarrow [0, \infty)$ es llamada una función clase $\mathcal{KL}$ si el primer argumento de la función $\beta(s, \cdot)$ es clase $\mathcal{K}$, y el segundo argumento de la función $\beta(\cdot, t)$ es estrictamente decreciente y $\lim_{t\rightarrow\infty} \beta(\cdot, t) = 0$.*

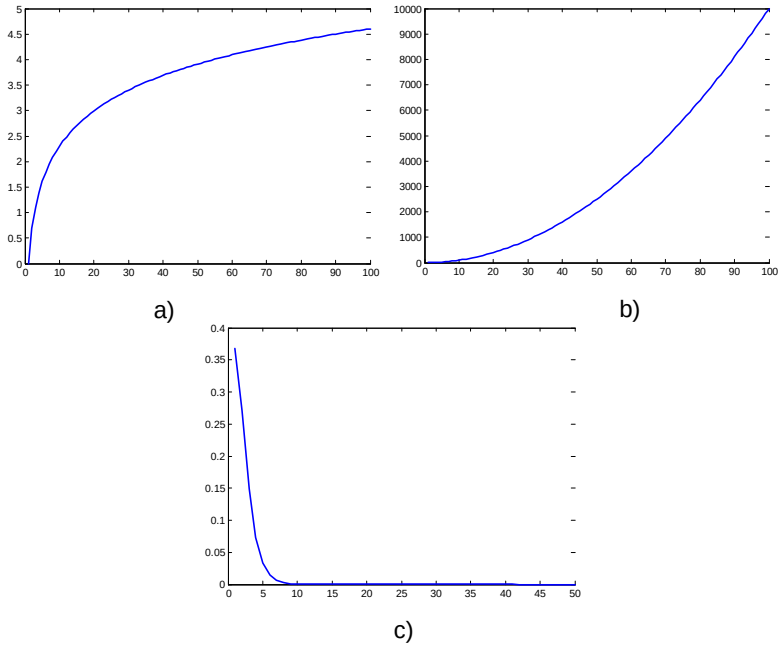

Figura 6.5: Tipos de funciones: a) clase k, b) clase k-infinito, c) clase kl

La ODE (**??**) es asintóticamente estable globalmente si existe una función $\beta \in \mathcal{KL}$ tal que:

$$\|\varphi(t,x)\| \leq \beta(\|x\|,t)$$

permanece para todo $x \in \mathbb{R}^n$, $t \geq 0$.

La comparación de funciones entre la solución de (6.7) y la función β de clase $\mathcal{KL}$ se muestra en la figura 6.6, esto mantiene la estabilidad asintótica globalmente de (6.7).

Una vez que se conoce el segundo método de Lyapunov, se introduce ahora las definiciones y conceptos de la estabilidad entrada-estado (ISS), el cuál es una extención de la estabilidad asintótica globalmente del sistema (6.7), sea el sistema no lineal:

$$\dot{x}(t) = f(x(t), u(t)) \tag{6.8}$$

donde $u(t) \in \mathbb{R}^m$ es la entrada ó perturbación del sistema y la salida son los estados $x \in \mathbb{R}^n$.

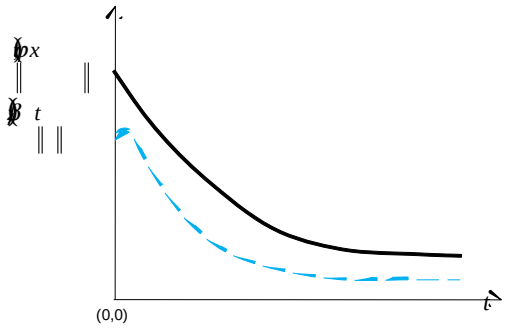

Figura 6.6: asintóticamente estable globalmente mediante comparación de funciones

Para los valores iniciales $x_0 \in \mathbb{R}^n$ con el tiempo inicial $t = 0$, $u(t)$ debe ser medible y acotada, i.e. $u(t) \in L_\infty$. La solución de (6.8) está dada por:

$$\varphi(t, x, u) \tag{6.9}$$

ISS requiere que el sistema con entrada permanezca GAS sobre un término de error dependiendo del tamaño de la perturbación $u(t)$ medida con:

$$\|u\|_\infty := \underset{t\geq 0}{ess\,\sup} \|u(t)\|$$

El sistema es llamado ISS, si existe una función $\beta \in \mathcal{KL}$ y una función $\gamma \in \mathcal{K}_\infty$ tal que para todos los valores iniciales de x, todas las perturbaciones u en todo el tiempo $t \geq 0$ la siguiente desigualdad permanece:

$$\|\varphi(t, x, u)\| \leq \text{máx}\left[\beta\left(\|x\|, t\right), \gamma\left(\|u\|_\infty\right)\right] \tag{6.10}$$

La figura 6.7, muestra graficamente la ecuación (6.10).

Ya que la solucion de (6.8) está dada por $\varphi(t, x, u)$, esta depende de $u(\tau)$ para $\tau \in [0, t]$ se puede deducir una desigualdad mas fuerte:

$$\|\varphi(t, x, u)\| \leq \text{máx}\left[\beta\left(\|x\|, t\right), \gamma\left(\left\|u|_{[0,t]}\right\|_\infty\right)\right]$$

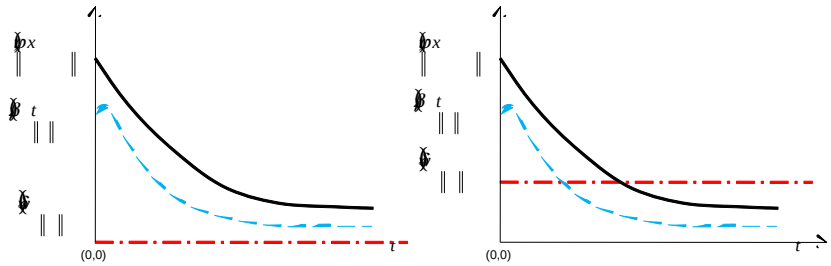

Figura 6.7: Estabilidad ISS

donde $u|_{[0,t]}(\tau) := \begin{cases} u(\tau), \tau \in [0,t] \\ 0, \tau \notin [0,t] \end{cases}$. Observando a β y γ separadamente, se identifican dos componentes:

GAS: asintóticamente estable globalmente para $u = 0$.

$$\|\varphi(t,x,0)\| \leq \beta(\|x\|,t)$$

Ganancia asintótica: La solución están acotados en última instancia por $\gamma(\|w\|_\infty)$

$$\lim_{t\to\infty} \sup \|\varphi(t,x,u)\| \leq \gamma(\|u\|_\infty)$$

es un hecho que es $ISS \iff GAS$ y ganancia asintótica ("⇐"no es tan trivial).

Si se considera un sistema lineal

$$\dot{x}(t) = Ax(t) + Bu(t)$$

con matrices A y B de dimensiones apropiadas.

Hecho 1: $GAS \Leftrightarrow A$ es Hurwitz, i.e. $\max_{\lambda \text{ eigenvalor de } A} Re(\lambda) < 0 \Leftrightarrow \left\|e^{At}\right\| \leq Ce^{-\sigma t}$, con $\sigma > 0$.

Hecho 2: $\varphi(t,x,u) = e^{At}x + \int_0^t e^{A(t-s)}Bu(s)\,ds \Leftrightarrow \|\varphi(t,x,u)\| \leq \beta(\|x\|,t) + \gamma(\|u\|_\infty)$

para $\beta(r,t) = Ce^{-\sigma t}r$ y $\gamma(r) = \|B\| \int_0^\infty \left\|e^{As}\right\| dsr \to ISS$.

Para sistemas lineales $GAS \Leftrightarrow ISS$.

Para sistemas no lineales esto no se cumple: considere la ecuación:

$$\dot{x} = u - sat\left(x\right)$$

donde $sat\left(x\right) = \begin{cases} 1, x > 1 \\ x, x \in [-1, 1] \\ -1, < -1 \end{cases}$. El sistema es GAS con entrada 0, pero para $u\left(t\right) = 2$ y $x = 1$ se obtiene la solución:

$$\varphi\left(t, x, u\right) = 1 + t$$

el cuál no es acotado, y la ISS no permanece. la ISS es mas fuerte que GAS.

6.4.1. Función de Lyapunov

La GAS puede ser caracterizada mediante una función de Lyapunov. Una ODE es GAS si y solamente si existe una función suave de Lyapunov $V : \mathbb{R}^n \to \mathbb{R}$ y las funciones $\alpha_1, \alpha_2 \in K_\infty$, $\alpha_3 \in \mathcal{K}$ tal que:

$$\alpha_1\left(\|x\|\right) \leq V\left(x\right) \leq \alpha_2\left(\|x\|\right)$$

y

$$DV\left(x\right) f\left(x\right) \leq -\alpha_3\left(\|x\|\right)$$

permanece para todo $x \in \mathbb{R}^n$. Donde $\frac{d}{dx} V\left(\varphi\left(t, x\right)\right) = DV\left(\varphi\left(t, x\right)\right) f\left(\varphi\left(t, x\right)\right) \leq -\alpha_3\left(\|\varphi\left(t, x\right)\|\right)$ implica que $V\left(\varphi\left(t, x\right)\right)$ es estrictamente decreciente y que tiende a 0.

Un sistema es ISS si y solo si existe una función ISS de Lyapunov $V : \mathbb{R}^n \to \mathbb{R}$ y las funciones $\alpha_1, \alpha_2, \alpha_4 \in K_\infty$, $\alpha_3 \in \mathcal{K}$ tal que:

$$\alpha_1\left(\|x\|\right) \leq V\left(x\right) \leq \alpha_2\left(\|x\|\right)$$

y $\|u\| \leq \alpha_4\left(\|x\|\right) \Rightarrow$

$$DV\left(x\right) f\left(x, u\right) \leq -\alpha_3\left(\|x\|\right)$$

permanece para todo $x \in \mathbb{R}^n, u \in \mathbb{R}^m$. De manera resumida podemos decir que:

- ISS es una generalización de GAS para sistemas lineales perturbados
- ISS combina GAS con entrada 0 y ganancia asintótica
- Para sistemas lineales ISS es equivalente a GAS con entrada 0, pero no se cumple para sistemas lineales
- ISS puede ser caracterizado mediante las funciones ISS de Lyapunov (si y solo si)

Todas estas características se extienden para sistemas discretos.