

Capítulo 4 Diseño de un control híbrido neurodifuso para una unidad turbogás

Chapter 4 Design of a hybrid neuro-fuzzy system for a gas turbine

VILLAGRAN-VILLEGAS, Luz Yazmin†*, PATIÑO-ORTIZ, Miguel, HERNÁNDEZ-GÓMEZ, Luis Héctor y BELTRÁN-FERNÁNDEZ, Juan Alfonso

Instituto Politécnico Nacional. Unidad Profesional “Adolfo López Mateos”- Zacatenco, Gustavo A. Madero, Mexico D.F.

Universidad Veracruzana. Facultad de Ingeniería Mecánica Eléctrica. Poza Rica Veracruz, México.

ID 1^{er} Autor: *Luz Yazmin, Villagran-Villegas* / **ORC ID:** 0000-0003-3860-2923, **CVU CONACYT ID:** 96365

ID 1^{er} Coautor: *Miguel, Patiño-Ortiz* / **ORC ID:** 0000-0002-5630-8077, **CVU CONACYT ID:** 167388

ID 2^{do} Coautor: *Luis Héctor, Hernández-Gómez* / **ORC ID:** 0000-0003-2573-9672, **CVU CONACYT ID:** 5107

ID 3^{er} Coautor: *Juan Alfonso, Beltrán-Fernández* / **ORC ID:** 0000-0002-3527-1754, **CVU CONACYT ID:** 38335

DOI: 10.35429/H.2019.2.50.68

L. Villagran, M. Patiño, L. Hernández y J. Beltrán

yvillagran@uv.mx

A. Marroquín, J. Olivares, P. Diaz y L. Cruz. (Dir.) La invención y las mujeres en Mexico. Handbooks-©ECORFAN-Mexico, Querétaro, 2019.

Resumen

En el Instituto Nacional de Electricidad y Energías Limpias se han desarrollado e implantado Sistemas de Control Distribuido para Centrales de Ciclo Combinado, en los cuales se han detectado varias necesidades y oportunidades de desarrollo tecnológico. Una de las más importantes es el mejoramiento de estrategias de control de velocidad, potencia y temperatura de las unidades turbogás, basadas en algoritmos de control PI convencional para obtener una operación más confiable y rentable, ya que las unidades turbogás demandan requerimientos más estrictos a los sistemas de control; debido a que las unidades turbogás se caracterizan por operar a temperaturas, presiones y velocidades relativamente más altas que las de cualquier otro tipo de unidades. Una alternativa atractiva para mejorar las estrategias de control de las unidades turbogás es el empleo de técnicas de control basadas en lógica difusa, las cuales podrían superar algunas de las desventajas del control moderno y podrían satisfacer requerimientos más sofisticados que con el control convencional. La introducción de controladores difusos usualmente está acompañada de una disminución del desempeño y la aparición de cambios no deseados en la respuesta del proceso durante el ajuste de los parámetros del controlador, lo cual es inadmisibles en unidades turbogás de alto riesgo. En esta investigación se atiende este último problema presentando el diseño y desarrollo de un controlador tipo PI que combina las técnicas de lógica difusa y redes neuronales en un sistema híbrido para el control de velocidad de una unidad turbogás que puede ser aplicado sin impactar negativamente el desempeño y la respuesta del proceso, y que posteriormente puede usarse para mejorarlos. El controlador PI neurodifuso emula a un control PI convencional y su diseño se realiza mediante aprendizaje neuronal supervisado a partir de un conjunto de patrones de entrada-salida determinados por el controlador convencional. El desarrollo del controlador PI neurodifuso se llevó a cabo en dos etapas. En la etapa inicial, el conjunto de patrones de entrenamiento se genera a partir del mapeo estático definido por la estructura y los parámetros del controlador PI convencional. El desempeño del controlador neurodifuso fue verificado mediante una serie de experimentos de simulación con un proceso de primer orden con retardo. Los resultados obtenidos demuestran la equivalencia entre el control neurodifuso y el control convencional, por lo cual pueden ser utilizados indistintamente en la misma aplicación. Con la técnica anterior de diseño se necesita conocer con exactitud la estructura del controlador y sus parámetros, sin embargo, en una situación real estos requerimientos dificultan la aplicación de este método, razón por la cual, en una segunda etapa, se optó por generar el conjunto de patrones de entrada-salida a partir de su medición directa durante el arranque de una unidad turbogás. El desempeño del controlador neurodifuso obtenido fue verificado mediante experimentos de simulación del arranque de una unidad turbogás con el modelo de una unidad GE5001 de 56 MW, demostrando la equivalencia de ambos controladores. Con los resultados obtenidos se puede concluir que el control neurodifuso puede sustituir de manera directa al control convencional sin presentar cambios indeseables en la respuesta del sistema. Además, se tiene la ventaja de que no es necesario conocer la estructura y los parámetros del control convencional a sustituir, facilitando su aplicación práctica para el control de velocidad de una unidad turbogás. El mejoramiento del desempeño empleando el controlador PI neurodifuso es un problema adicional que se planteó como un problema de optimización, cuya solución permite mejorar características específicas de la operación durante el arranque de la unidad turbogás. La solución consistió en encontrar valores para los factores de escalamiento de las entradas y salidas del controlador que minimicen un índice de desempeño (función objetivo) arbitrario. El índice de desempeño se escoge de acuerdo a las características que se desean mejorar, por ejemplo el error de seguimiento de la curva de arranque, o el esfuerzo de control. Los resultados obtenidos muestran la mejora en el desempeño de la unidad turbogás, en las características consideradas en los índices de desempeño, lo cual demuestra la efectividad de la estrategia propuesta. Los resultados de esta investigación, para dar soluciones a los problemas de diseño de controladores neurodifusos y para mejorar el desempeño de las unidades turbogás, constituyen una aportación valiosa al conocimiento para lograr una operación más segura y rentable de las unidades turbogás a corto plazo.

Control neurodifuso, Turbina de gas, Redes neuronales, Control PI

Abstract

Several needs and opportunities have been detected during the development of Distributed Control Systems for Combined Cycle Power Plants at the National Institute of Electricity and Clean Energies. Among the most important is the improvement of the speed, power and temperature control strategies for turbogas units, currently based on conventional PI algorithms, to achieve safer and more profitable operation. To a great extent, this is because turbogas units pose a larger number of tighter requirements to the control system since these units operate at relatively higher temperature, speed and pressure than other power units. An attractive alternative to improve the control strategies at turbogas units consists on the deployment of control techniques based on fuzzy logic, which may supersede some of the disadvantages of modern control approaches and satisfy more sophisticated requirements than with conventional control. Nevertheless, deployment of fuzzy control is usually accompanied with a performance reduction and undesired upsets on the process response during the parameter tuning stage, which is unacceptable for high-risk turbogas units. This thesis research deals with the latest problem through the design and development of a PI like controller that merges fuzzy logic and neural network techniques into a hybrid system for speed control of a turbogas unit. The proposed controller may be applied without negatively impacting the process performance and response; later it could be used to improve them. The proposed neurofuzzy PI controller emulates a conventional PI controller; it is designed through supervised neural learning from a set of input-output patterns defined by the conventional controller. The neurofuzzy PI development is carried out in two stages. In the initial stage, the set of training patterns is generated through the static mapping defined by the structure and parameters of the conventional PI controller. Performance of this neurofuzzy PI controller was verified through simulation experiments with a first order with time-delay process model. Results show the equality of both the neurofuzzy controller and the conventional controller, thus they may be used interchangeably in the same application. In the former design approach, it is necessary to precisely know the controller structure and its parameters; nevertheless, these requirements may prevent application in a real world situation. Therefore, in a second development stage, it was decided to generate the input-output training patterns through direct measurement during the turbogas unit start-up. Performance of the resultant neurofuzzy controller was verified through start-up simulation experiments with the mathematical model of a 25 MW GE-5001 turbogas unit. Results show the equivalence of both controllers. With these results, it is concluded that the neurofuzzy controller can directly substitute the conventional controller without causing any undesirable upsets on the plant response. In addition, it is not necessary to know the structure and parameters of the conventional controller to be substituted, thus easing its application for speed control of a turbogas unit. The results of this research, in the form of solutions to the design of neurofuzzy controllers with a guarantee on their performance, constitute a worthy contribution towards achieving safer and more profitable operation of a turbogas unit in the short term.

Neuro-fuzzy control, Gas turbine, Neural network, PI controller

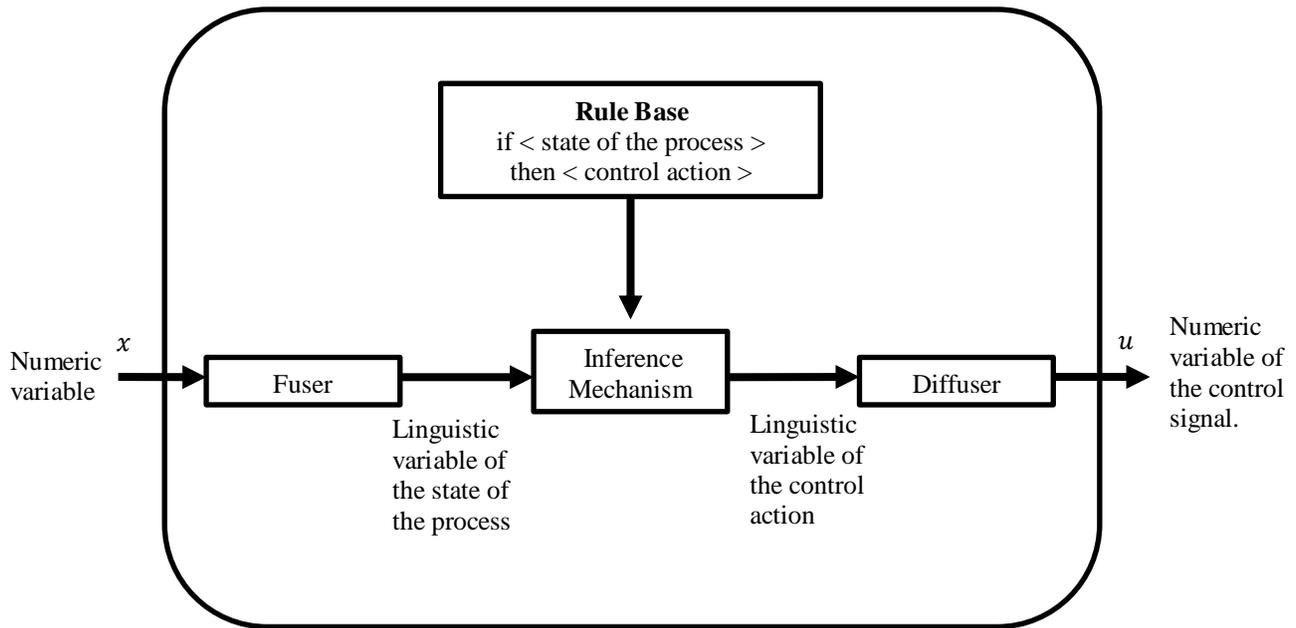
1. Introducción

Una alternativa atractiva para mejorar las estrategias de control de las unidades turbogás es el empleo de técnicas de control basadas en lógica difusa. Estas técnicas resultan atractivas porque no requieren un modelo matemático del proceso como las técnicas de control moderno y cuentan con medios adicionales para satisfacer requerimientos de operación más sofisticados que con el control convencional.

Un controlador difuso (CD) genera la señal de control mediante un proceso de inferencia que emula el razonamiento humano, utilizando reglas de procedimiento y conocimiento sobre el estado del proceso. La parte medular de un controlador difuso es un sistema de inferencia difuso formado por cuatro componentes principales: fusificador, base de reglas, mecanismo de inferencia y defusificador (Figura 4.1). Cualitativamente, la operación del controlador es la siguiente: El fusificador transforma las variables numéricas de entrada al controlador en variables lingüísticas que describen el estado del proceso. La base de reglas contiene el conocimiento de los procedimientos de control para la planta en cuestión en forma de reglas de procedimiento del tipo: si <estado del proceso> entonces <acción de control>, a partir de los valores de las variables lingüísticas de entrada, el mecanismo de inferencia determina que reglas son activadas, las evalúa y obtiene conclusiones respecto a las acciones de control a ejercer sobre la planta, emulando los mecanismos de razonamiento humano.

El desfusificador transforma las acciones de control expresadas como variables lingüísticas a un valor numérico [Wang, 1997].

Figura 4.1 Sistema de inferencia difuso



Los controladores difusos directos se caracterizan por encontrarse dentro de un lazo de control y son aplicados en lugar de un controlador convencional. Debido al tipo de aplicación en esta investigación resultan de interés los controladores difusos que emulan a los controladores PID, y de manera más específica interesa el diseño de los sistemas de inferencia de los controladores difusos que sean capaces de sustituir a un controlador PI convencional. [Sánchez, 2019].

En general, el diseño de un sistema de inferencia difuso no es una tarea trivial, ya que se necesitan tomar muchas decisiones, entre las que destacan las siguientes: seleccionar el método de fusificación, seleccionar el método de inferencia, seleccionar el método de desfusificación, determinar el universo de discurso, determinar los factores de escalamiento, determinar las funciones de membresía de la entrada del sistema, determinar las funciones de membresía de la salida del sistema, determinar la base de reglas, seleccionar el conectivo entre premisas, determinar los parámetros y el contenido de las reglas.

Los problemas de interés en este trabajo son para determinar la estructura y el ajuste de los parámetros del sistema de inferencia de un controlador difuso. El problema del ajuste o sintonización consiste en determinar el valor de los parámetros que definen a las funciones de membresía de los antecedentes y de los consecuentes de las reglas. Tradicionalmente, el ajuste de los parámetros de un sistema de inferencia difuso es realizado mediante un proceso de ensayo y error. Este proceso de sintonización manual es muy laborioso y susceptible de tener errores, lo cual podría causar respuestas indeseables en el sistema; degradando significativamente su desempeño. Esta situación es inadmisibles en procesos de alto riesgo como lo es una UTG, por esta razón es deseable contar con un método que permita realizar el proceso de sintonización en forma automática.

Los sistemas neurodifusos pueden ser empleados en la corrección de errores, clasificación de datos, predicción de valores, etc. En esta investigación se muestra un sistema neurodifuso que permite automatizar el diseño de un sistema de inferencia difuso. En un sistema neurodifuso (SND) se fusionan conceptos de sistemas difusos y redes neuronales para obtener un solo sistema que presente las ventajas de ambas técnicas. El interés de las redes neuronales es contar con su capacidad de aprendizaje, optimización, y una estructura que garantice una degradación paulatina del desempeño del sistema en caso de falla. Así mismo, los sistemas de inferencia difusos permiten contar con la capacidad de operar bajo incertidumbre con conocimiento aproximado, procesar información con reglas de inferencia del tipo si-entonces, y tener facilidad para incorporar e interpretar el conocimiento de operadores expertos para el control de procesos.

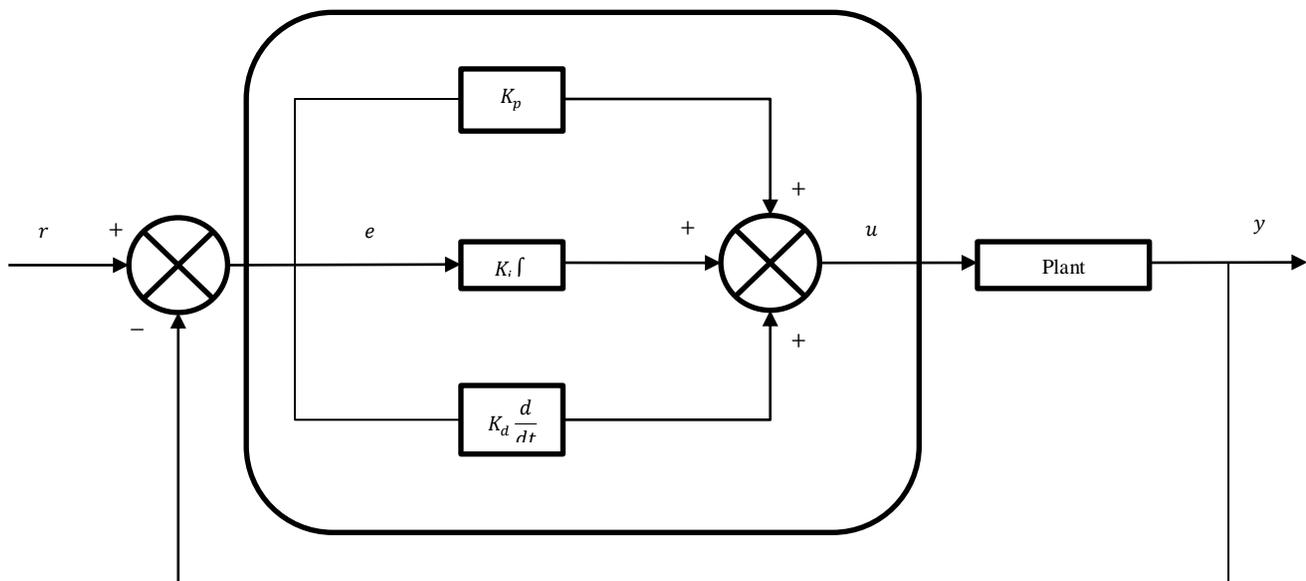
En la literatura se encuentran disponibles diversos paradigmas de sistemas neurodifusos, entre los más atractivos para ser aplicados al diseño de un controlador neurodifuso se encuentran los siguientes: FALCON (Fuzzy Adaptive Learning Control Network) propuesto por [Lin y Lee, 1991] que ha sido usado para estudiar estrategias de aprendizaje de estructura y parámetros; ARIC (Approximate Reasoning-based Intelligent Control) propuesto por [Berenji, 1992] que utiliza varias redes neuronales prealimentadas; GARIC (Generalized ARIC) propuesto para mejorar el método ARIC [Berenji y Khedkar, 1992]; ANFIS (Adaptive Network-based Fuzzy Inference System) propuesto por [Jang, 1993] para aproximar funciones usando sistemas difusos del tipo Sugeno; NEFCON (Neuro Fuzzy Control) capaz de aprender y optimizar las reglas básicas de un sistema difuso del tipo Mamdani [Nauck, 1994]; NNDFR (Neural Network Driven Fuzzy Reasoning) por [Takagi, 1991] basada en redes neuronales comunes que son estructuradas con técnicas de sistemas difusos; FuNe por [Halgamuge, 1994] basado en la arquitectura de una red neuronal prealimentada de cinco capas.

Entre los paradigmas neurodifusos mencionados se destacan ANFIS y NEFCON. ANFIS representa a un sistema de inferencia tipo Sugeno mediante una red neuronal prealimentada de cinco capas, cuyos parámetros son determinados por un procedimiento de retropropagación usando mínimos cuadrados. NEFCON representa aun sistema de inferencia difuso tipo Mamdani mediante una red tipo perceptrón de 3 capas y los parámetros son encontrados por retropropagación usando medidas de error difuso. En este proyecto se utiliza el paradigma ANFIS debido a su disponibilidad en software comercial.

1.1. Configuración del controlador PI neurodifuso

El diseño de la estructura del controlador PI neurodifuso parte de la estructura de un controlador PID (Figura 4.2).

Figura 4.2 Controlador PID Convencional



La señal de control del controlador PID convencional se calcula con:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} + u_o \quad (1.1)$$

En donde: $u(t)$ es la salida de la señal de control, $e(t)$ es señal de error, t es el tiempo, K_p es la ganancia proporcional, K_i es la ganancia integral y K_d es la ganancia derivativa.

Dado que la implementación del controlador se hará en un controlador digital, es necesario tener una expresión en tiempo discreto de la señal de control, con este fin la derivada y la integral del error se aproximan numéricamente como:

$$\frac{de(t)}{dt} = \frac{e(k) - e(k-1)}{T} \quad (1.2)$$

$$\int_0^t e(t)dt = T \sum_{i=0}^{k-1} e(i) \quad (1.3)$$

En donde k es el instante de muestreo y T es el periodo.

Sustituyendo (1.2) y (1.3) en (1.1) se obtiene la expresión:

$$u(k) = K_p e(k) + K_i T \sum_{i=0}^{k-1} e(i) + K_d \frac{1}{T} (e(k) - e(k-1)) + u_0 \quad (1.4)$$

La configuración tipo posición del controlador PID en (1.4) proporciona la acción de control directamente. Sin embargo, para la programación en computadora del algoritmo de control se prefiere una expresión recurrente de la forma:

$$u(k) = u(k-1) + \Delta u(k) \quad (1.5)$$

En donde $\Delta u(k)$ es el cambio en la señal de control entre dos instantes de muestreo consecutivos.

Rescribiendo (1.5) como:

$$\Delta u(k) = u(k) - u(k-1) \quad (1.6)$$

y tomando en cuenta que de (1.4):

$$u(k-1) = K_p e(k-1) + K_i T \sum_{i=0}^{k-2} e(i) + K_d \frac{1}{T} (e(k-1) - e(k-2)) + u_0 \quad (1.7)$$

$$\Delta u(k) = K_p (e(k) - e(k-1)) + K_i T e(k-1) + \frac{K_d}{T} (e(k) - 2e(k-1) + e(k-2)) \quad (1.8)$$

En conjunto las ecuaciones (1.8) y (1.5) constituyen una expresión alternativa para el controlador PID convencional en tiempo discreto denominada de tipo velocidad.

Para los objetivos de este trabajo se requiere un controlador PI, el cual puede obtenerse haciendo $K_d=0$ en (1.8):

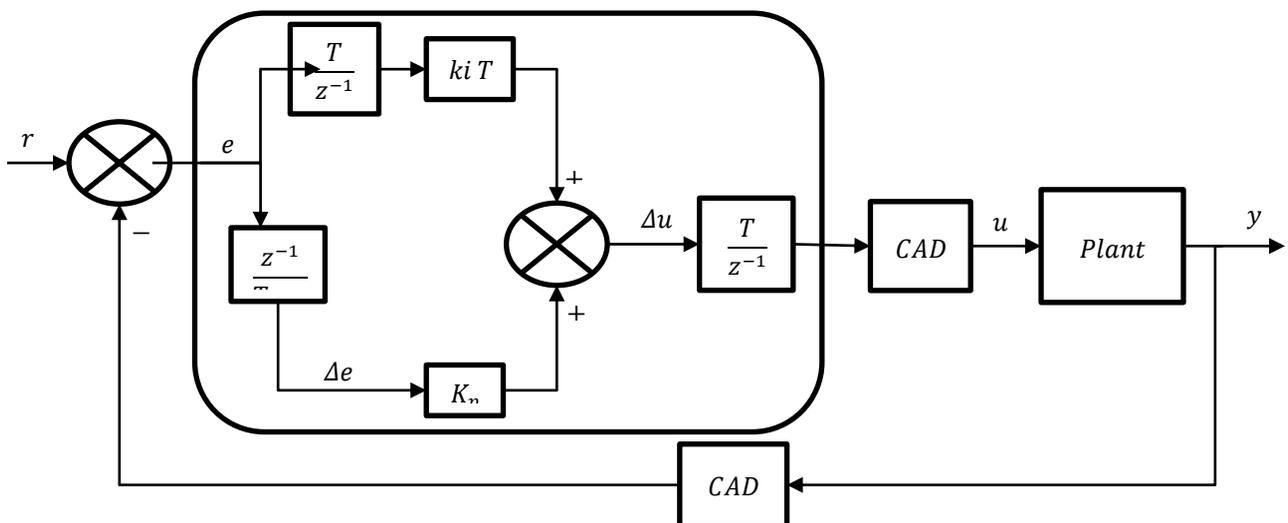
$$\Delta u(k) = K_p (e(k) - e(k-1)) + K_i T e(k-1) \quad (1.9)$$

ó en forma equivalente:

$$\Delta u(k) = K_p \Delta e(k) + K_i T e(k-1) \quad (1.10)$$

En donde $e(k)$, definida apropiadamente, es el cambio en la señal de error. Con base en esta última relación la estructura de un controlador PI convencional sería como se muestra en la Figura 4.3.

Figura 4.3 Configuración del controlador PI convencional digital



De la relación (1.10) puede verse que un controlador PI define un mapeo estático, $R^2 \rightarrow R$, del error anterior, $e(k-1)$, y el cambio del error actual, $\Delta e(k)$, a el cambio en la acción de control actual, $\Delta u(k)$. Este mapeo estático puede ser aproximado por un sistema de inferencia difuso de dos entradas y una salida, tal y como se muestra en la Figura 4.4, ya que los sistemas difusos son aproximadores universales que pueden aproximar cualquier relación estática no lineal entre sus entradas y salidas con cualquier nivel de precisión deseado [Wang, 1997]. De esta manera, la estructura del controlador PI convencional (Figura 4.3) define la estructura del controlador PI neurodifuso (Figura 4.5). Nótese que ambos controladores presentan una configuración tipo velocidad, esto es, primero calculan el cambio en la señal de control y después la señal de control.

Figura 4.4 Aproximación del mapeo Estático del controlador PI mediante un sistema difuso

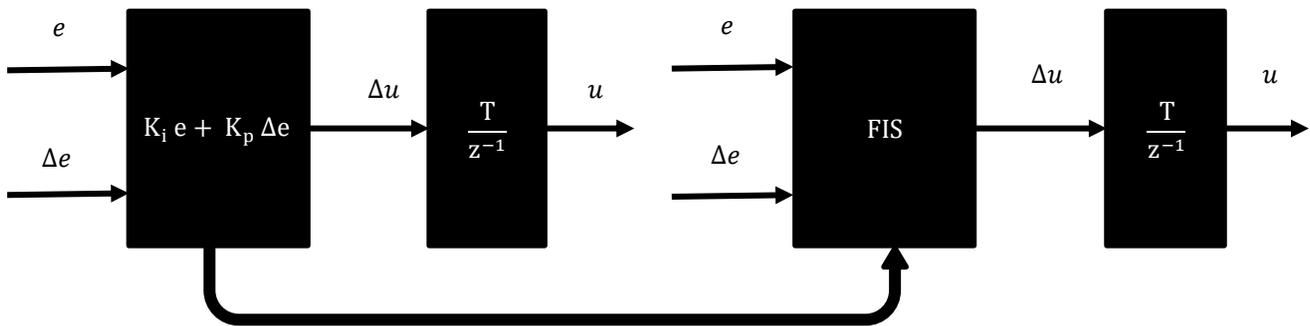
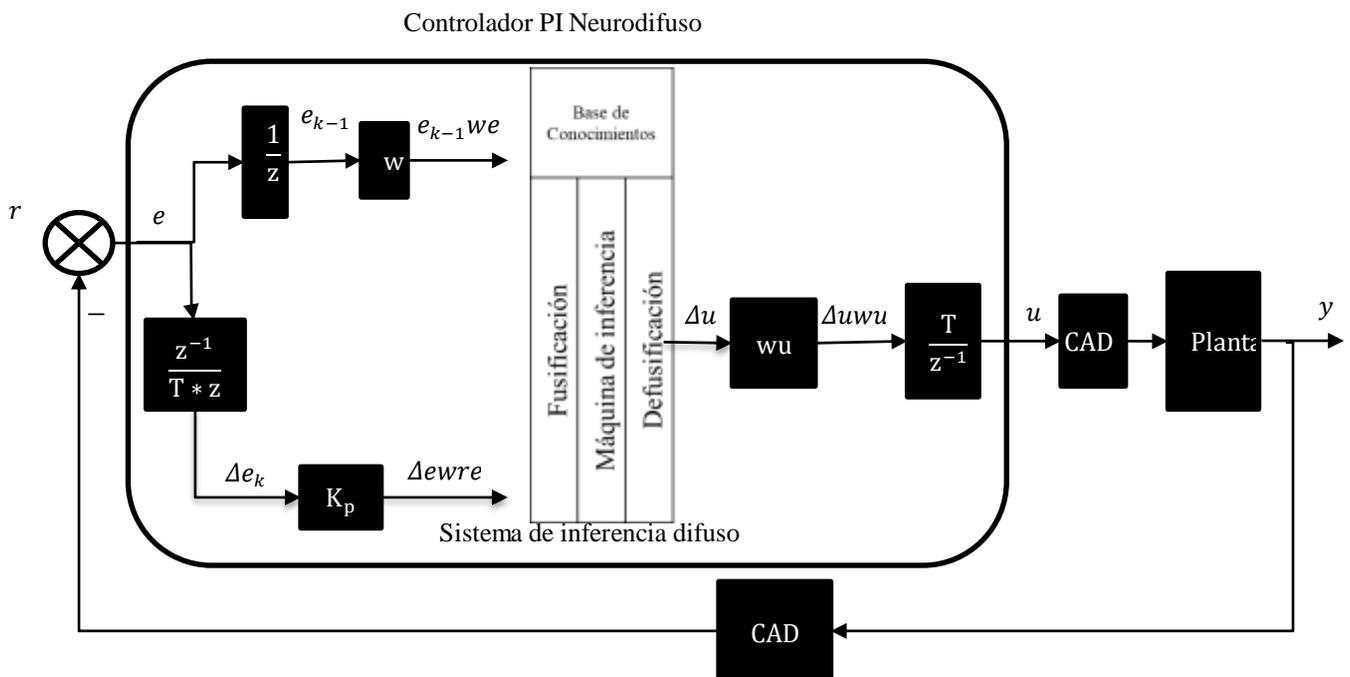


Figura 4.5 Configuración del controlador PI neurodifuso



1.2. Estructura del sistema de inferencia

Una vez que la estructura del controlador PI neurodifuso ha sido definida, el siguiente problema consiste en definir la estructura del sistema de inferencia difuso correspondiente, para lo cual es necesario adelantarse y considerar el problema de la determinación de los parámetros del sistema de inferencia difuso. El ajuste de los parámetros de los diversos componentes de un sistema difuso para aproximar el mapeo estático del controlador PI convencional, con una precisión razonablemente buena, no es una tarea trivial que puede ser ejecutada por ensayo y error; la respuesta de la unidad turbogás puede deteriorarse o pueden ocurrir trastornos importantes en la operación, siendo ambas situaciones inaceptables para las unidades turbogás, que son consideradas como procesos de alto riesgo. A este respecto, los métodos de aprendizaje con redes neuronales pueden ser utilizados ventajosamente para llevar a cabo esta tarea. Para ello, el sistema difuso debe ser representado como una red neuronal, de ahí el nombre de controlador PI neurodifuso.

Entonces, la determinación de los parámetros del sistema difuso es equivalente a encontrar los parámetros de la red neuronal, los cuales son calculados por el algoritmo de aprendizaje neuronal.

En este trabajo, los parámetros del sistema difuso son calculados con el método de aprendizaje neuronal ANFIS [Jang, 1993] que se encuentra comercialmente disponible como una librería del paquete de programación Matlab. En el método ANFIS se requiere que el sistema difuso sea del tipo TSK (Takagi- Sugeno-Kan), con reglas de inferencia de la forma:

$$\text{SI } \langle e(k-1) \text{ es } LE_r \rangle \text{ y } \langle \Delta e(k) \text{ es } L\Delta E_r \rangle \text{ ENTONCES } \Delta u_r(k) = p_r e(k-1) + q_r \Delta e(k) + k_r \quad (1.11)$$

En donde $r = 1, 2, \dots, R$ es el número de regla, LE_r y $L\Delta E_r$ son los valores lingüísticos de las señales de entrada e y Δe respectivamente, en la r -ésima regla, Δu es la contribución de la r -ésima regla a la salida total del sistema difuso, y p_r , q_r y k_r son los coeficientes del consecuente de la r -ésima regla.

Para un patrón de entrada determinado $[e(k-1)\Delta e(k)]$, la salida del sistema difuso tipo TSK, $\Delta u(k)$, está dada por:

$$\Delta u(k) = \frac{\sum_{r=1}^R w^r \Delta u^r(k)}{\sum_{r=1}^R w^r} \quad (1.12)$$

En donde w^r , para $r = 1, 2, \dots, R$, es el grado de cumplimiento o peso de disparo de la r -ésima regla. Para cada regla, su grado de cumplimiento es calculado como el producto de los valores de pertenencia de las entradas:

$$w^r = \mu_{LE^r}(e(k-1))\mu_{L\Delta E^r}(\Delta e(k)) \quad (1.13)$$

$$\mu_{LE^r}(\cdot) \text{ y } \mu_{L\Delta E^r}(\cdot) \quad (1.14)$$

ésima regla. En preparación para la representación del sistema difuso TSK mediante una red neuronal, la salida (1.12) puede escribirse como:

$$\Delta u(k) = \sum_{r=1}^R \left(\frac{w^r}{\sum_{r=1}^R w^r} \right) \Delta u^r = \sum_{r=1}^R \bar{w}^r \Delta u^r = \sum_{r=1}^R \bar{\Delta u}^r \quad (1.15)$$

En donde \bar{w}^r , para $r = 1, 2, \dots, R$, son los llamados grados relativos (normalizados) de cumplimiento de las reglas:

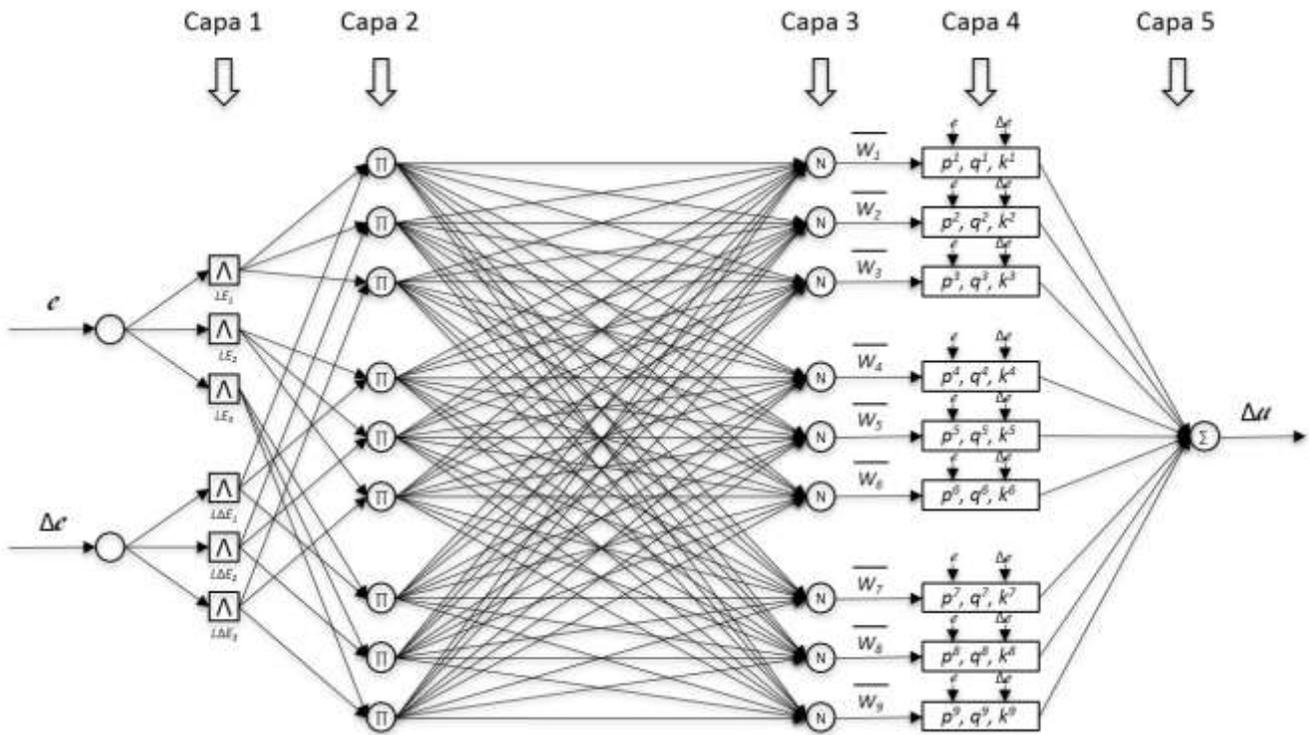
$$\bar{w}^r = \frac{w^r}{\sum_{r=1}^R w^r} \quad (1.16)$$

Y los $\bar{\Delta u}^r$, para $r = 1, 2, \dots, R$, pueden ser equivalentemente llamados consecuentes normalizados:

$$\bar{\Delta u}^r = \bar{w}^r \Delta u^r \quad (1.17)$$

El sistema de inferencia difuso TSK es representado mediante una red neuronal prealimentada (feedforward) de cinco capas, como se muestra en la Figura 4.6 para el caso en que cada entrada tiene tres funciones de pertenencia (valores lingüísticos). A grandes rasgos, la red neuronal fusifica las entradas en la capa 1, implementa una regla de inferencia por cada fila horizontal en las tres capas intermedias, y compone el cambio en la señal de control en la última capa.

Figura 4.6 Representación del sistema de inferencia difuso como una red neuronal



En forma más precisa y para el caso de la Figura 4.6, la operación de la red neuronal que representa al sistema difuso del controlador PI neurodifuso es la siguiente:

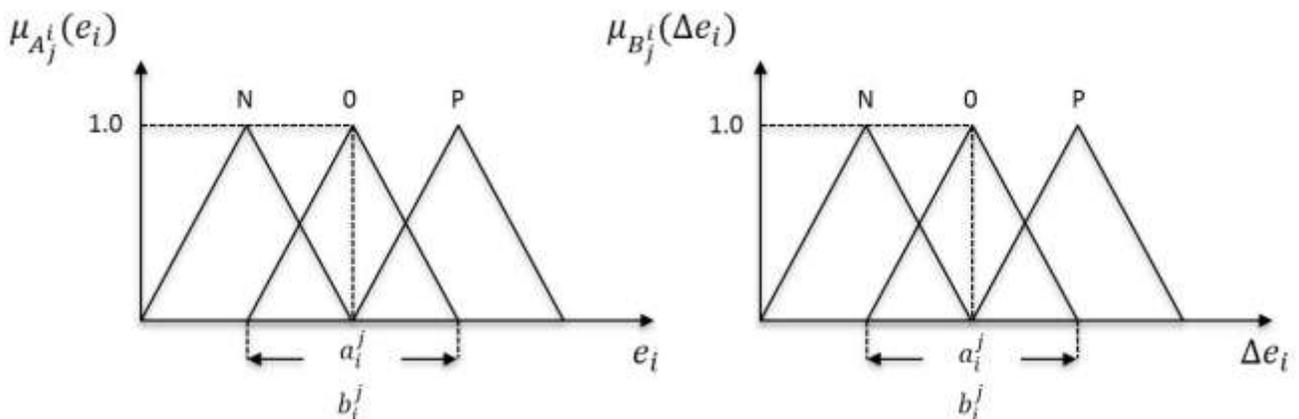
Capa 1. En esta capa, en cada neurona, se fusifica la señal de entrada (1.14) utilizando una función de pertenencia triangular (Figura 4.7):

$$O_{1,i} = \mu_{LE_i}(\Delta e(k)) = 1 - \frac{2|e(k-1) - a_i|}{b_i} \quad \text{para } i = 1, 2, 3 \quad (1.18)$$

$$\mu_{\Delta LE_1}(\Delta e(k)) = 1 - \frac{2|e(k-1) - a_i|}{b_i} \quad \text{para } i = 4, 5, 6 \quad (1.19)$$

En donde $O_{1,i}$ denota la salida de la i -ésima neurona de la Capa 1, LE_i , para $i=1, 2, 3$, y ΔLE_i , para $i = 4, 5, 6$, son los términos lingüísticos (N: negativo, 0: cero, y P: positivo) implementados en cada neurona, y a_i y b_i denotan el centro y el ancho de la base del triángulo correspondiente. Los parámetros a_i y b_i serán ajustados por el algoritmo de aprendizaje y por tanto las neuronas de la Capa 1 son consideradas como nodos adaptables.

Figura 4.7 Funciones de pertenencia triangulares



Capa 2. Las neuronas de esta capa calculan el grado de cumplimiento de las reglas (1.13), una regla por neurona, utilizando el producto algebraico para la norma T:

$$O_{2,r} = w^r = \mu_{LE^r}(e(k-1))\mu_{L\Delta E^r}(\Delta e(k)) \quad (1.20)$$

En donde $O_{2,r}$ es la salida de la r -ésima neurona, $r = 1, 2, \dots, 9$, es el número de regla. Nótese que el número de neurona y el número de regla coinciden. Por lo tanto los pesos son:

$$\begin{aligned} w_1 &= \mu_{LE1}(e_{(k-1)})\mu_{L\Delta E1}(\Delta e(k)) & w_2 &= \mu_{LE1}(e_{(k-1)})\mu_{L\Delta E2}(\Delta e(k)) & w_3 &= \mu_{LE1}(e_{(k-1)})\mu_{L\Delta E3}(\Delta e(k)) \\ w_4 &= \mu_{LE2}(e_{(k-1)})\mu_{L\Delta E1}(\Delta e(k)) & w_5 &= \mu_{LE2}(e_{(k-1)})\mu_{L\Delta E2}(\Delta e(k)) & w_6 &= \mu_{LE2}(e_{(k-1)})\mu_{L\Delta E3}(\Delta e(k)) \\ w_7 &= \mu_{LE3}(e_{(k-1)})\mu_{L\Delta E1}(\Delta e(k)) & w_8 &= \mu_{LE3}(e_{(k-1)})\mu_{L\Delta E2}(\Delta e(k)) & w_9 &= \mu_{LE3}(e_{(k-1)})\mu_{L\Delta E3}(\Delta e(k)) \end{aligned} \quad (1.21)$$

Las neuronas de esta capa son consideradas nodos fijos, etiquetados con Π , ya que no tienen parámetros que ajustar.

Capa 3. Las neuronas de esta capa calculan el grado relativo de cumplimiento (1.16) de las reglas, una regla por neurona, esto es, la r -ésima neurona calcula el grado relativo de cumplimiento de la r -ésima regla:

$$O_{3,r} = \bar{w}^r = \frac{w^r}{\sum_{r=1}^r w^r} \quad (1.22)$$

Las neuronas de esta capa también son consideradas como nodos fijos.

Capa 4. Las neuronas en esta capa calculan los consecuentes normalizados (1.17) para cada regla, una neurona por consecuente, esto es, la r -ésima neurona calcula el consecuente normalizado de la r -ésima regla:

$$O_{4,r} = \bar{\Delta u}^r(k) = \bar{w}^r \Delta u^r(k) = \bar{w}^r (p^r e(k-1) + q^r \Delta e(k) + k^r) \quad (1.23)$$

En donde p^r , q^r y k^r son los coeficientes del consecuente de la r -ésima regla, los cuales serán ajustados por el algoritmo de aprendizaje y por tanto las neuronas de la Capa 4 son consideradas como nodos adaptables.

Capa 5. El único nodo presente en esta capa es un nodo fijo, denotado por Σ , que calcula la salida del sistema difuso (1.15), o cambio en la señal de control, como la suma de todos los consecuentes normalizados:

$$O_5 = \Delta u(k) = \sum_{r=1}^R \bar{\Delta u}^r(k) \quad (1.24)$$

1.3. Procedimiento de sintonización del sistema difuso

El sistema difuso debió ser representado como una red neuronal con lo cual la determinación de los parámetros del sistema difuso es equivalente a encontrar los parámetros de la red neuronal que deben ser calculados con el método de aprendizaje neuronal ANFIS [Jang,1993]. Además, las reglas de inferencia son del tipo TSK como en (1.11) que se repite aquí por conveniencia:

$$\text{SI } \langle e(k-1) \text{ es } LE^r \rangle \text{ y } \langle \Delta e(k) \text{ es } L\Delta E^r \rangle \text{ ENTONCES } \Delta u^r(k) = p^r e(k-1) + q^r \Delta e(k) + k^r \quad (1.25)$$

Sin pérdida de generalidad, el problema de la sintonización del sistema difuso, suponiendo tres términos lingüísticos por entrada, puede definirse como sigue:

Dado un conjunto de M patrones de datos de entrada y salida $\{[e_1, \Delta e_1, \Delta u_1], [e_2, \Delta e_2, \Delta u_2], \dots, [e_M, \Delta e_M, \Delta u_M]\}$, generados por el controlador PI digital convencional, y un sistema difuso TSK definido por 9 reglas de la forma (1.25); el procedimiento ajusta los valores de los parámetros correspondientes a las funciones de membresía de las entradas del sistema difuso $\{[a_1, b_1], [a_2, b_2], \dots, [a_6, b_6]\}$, y a los coeficientes de los consecuentes de las reglas de inferencia $\{[p_1, q_1, k_1], [p_2, q_2, k_2], \dots, [p_9, q_9, k_9]\}$, para reproducir los patrones de datos de entrada y salida proporcionados.

El procedimiento de aprendizaje se lleva a cabo iterativamente, con dos fases por iteración. Primero, los patrones de entrada se propagan manteniendo constantes los parámetros de las funciones de pertenencia, y se calculan valores óptimos para los coeficientes de los consecuentes, usando un procedimiento de mínimos cuadrados estimados (Least Square Estimate - LSE). Segundo, los patrones de entrada se propagan nuevamente, manteniendo constantes los coeficientes de los consecuentes recién calculados, y entonces los parámetros de las funciones de transferencia se modifican por la regla de retropropagación. En lo que sigue, ambas fases son explicadas con más detalle.

Por una parte, los coeficientes de los consecuentes son calculados usando un procedimiento de mínimos cuadrados estimados (LSE) tomando en cuenta que cada patrón de entrada-salida se relaciona por:

$$\Delta u_m = \sum_{r=1}^9 \bar{w}^r (p^r e_m + q^r \Delta e_m + k^r) \quad (1.26)$$

Donde $m = 1, 2, \dots, M$ es el índice de patrones de entrada-salida. Usando vectores y considerando los M patrones de entrenamiento de entrada-salida se puede escribir:

$$\begin{bmatrix} \Delta u \\ \cdot \\ \cdot \\ \cdot \\ \Delta u_M \end{bmatrix} = \begin{bmatrix} \bar{w}^{-1} e_1 & \bar{w}^{-1} \Delta e_1 & \bar{w}^{-1} & \cdot & \cdot & \cdot & \bar{w}^{-9} e_1 & \bar{w}^{-9} \Delta e_1 & \bar{w}^{-9} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \bar{w}^{-1} e_M & \bar{w}^{-1} \Delta e_M & \bar{w}^{-1} & \cdot & \cdot & \cdot & \bar{w}^{-9} e_M & \bar{w}^{-9} \Delta e_M & \bar{w}^{-9} \end{bmatrix} = \begin{bmatrix} p^1 \\ q^1 \\ k^1 \\ \cdot \\ \cdot \\ \cdot \\ p^9 \\ q^9 \\ k^9 \end{bmatrix} \quad (1.27)$$

Que mediante definiciones apropiadas puede escribirse como:

$$U = XC \quad (1.28)$$

Donde U es $M \times 1$, X es $M \times (3) (9) = M \times 27$, y C es 27×1 . En general el problema de calcular los coeficientes en C es sobre determinado, es decir, $M > 27$. Una solución LSE para C puede ser calculada recursivamente usando: [Naucke,1997]

$$C_{i+1} = C_i + \psi_{i+1} x_{i+1} (u_{i+1}^T - x_{i+1}^T C_i) \quad (1.29)$$

$$\psi_{i+1} = \psi_i - \frac{\psi_i x_{i+1} x_{i+1}^T \psi_i}{1 + x_{i+1}^T \psi_i x_{i+1}} \quad (1.30)$$

Donde x_i es el i -ésimo vector renglón de la matriz X y u_i es el i -ésimo elemento del vector u , para $i=0, 1, 2, \dots, M-1$, Ψ es llamada la matriz de covarianza y T es la transpuesta. Las condiciones iniciales son $C_0 = 0$ y $\Psi_0 = \gamma I$, donde I es una matriz identidad de tamaño 27 y γ es un número positivo grande. Al final de las iteraciones, $C = CM$ habrá sido calculada usando toda la información disponible en los M patrones de entrada-salida.

Por otra parte, los cambios en los parámetros de las funciones de membresía se determinan por medio de retropropagación. Supongamos que z sea cualquiera de los parámetros a, b de cualquier función de membresía μ , y que E_{i0} sea la medida del error usual dada por el cuadrado de la diferencia entre la salida deseada Δu^* y la salida actual, Δu :

$$E_{i0} = \frac{1}{2} (\Delta u^* - \Delta u)^2 \quad (1.31)$$

Entonces, el cambio en los parámetros z , Δz , para una sola regla después de que un patrón ha sido propagado, está dado por:

$$\Delta z = -\sigma \frac{\partial E_{io}}{\partial z} \quad (1.32)$$

Donde

La aplicación sucesiva de la regla de la cadena a (1.31) a través de cada una de las capas de la red neuronal produce:

$$\begin{aligned} \Delta z &= -\sigma \frac{\partial E_{io}}{\partial z} \frac{\partial \Delta u}{\partial w^r} \frac{\partial \bar{w}^r}{\partial w^r} \frac{\partial w^r}{\partial \mu} \frac{\partial \mu}{\partial z} \\ &= -\sigma (\Delta u^* - \Delta u) \Delta u^r \frac{\bar{w}^r (1 - \bar{w}^r) w^r}{w^r} \frac{\delta \mu}{\partial z} \\ &= \frac{\sigma}{\mu} \Delta u^r (\Delta u^* - \Delta u) \bar{w}^r (1 - \bar{w}^r) \frac{\delta \mu}{\partial z} \end{aligned} \quad (1.33)$$

$$\frac{\delta u}{\delta a} = \frac{2 \operatorname{sgn}(e^{(k-1)} - a_i)}{b_i \mu_i e^{(k-1)}} \quad (1.34)$$

$$\frac{\delta \mu}{\delta b} = \frac{1 - \mu e^{(k-1)}}{b_i \mu_i e^{(k-1)}} \quad (1.35)$$

De esta manera, los cambios de los parámetros a y b, Δa y Δb , respectivamente, en una regla después de que el patrón ha sido propagado pueden calcularse sustituyendo (1.34) y (1.35) en (1.33).

$$\Delta a = -\frac{\sigma}{\mu} \Delta u^r (\Delta u^* - \Delta u) w^r (1 - \bar{w}^r) \frac{2 \operatorname{sgn}(e^{(k-1)} - a_i)}{b_i \mu_i e^{(k-1)}} \quad (1.36)$$

$$\Delta b = -\frac{\sigma}{\mu} \Delta u^r (\Delta u^* - \Delta u) w^r (1 - \bar{w}^r) \frac{1 - \mu e^{(k-1)}}{b_i \mu_i e^{(k-1)}} \quad (1.37)$$

1.4. Procedimiento de aprendizaje

Como se mencionó anteriormente, el procedimiento de aprendizaje se lleva a cabo iterativamente y consiste de los siguientes pasos:

1. Propagar todos los patrones del conjunto de aprendizaje y determinar los coeficientes de los consecuentes por medio del LSE iterativo en (1.29) y (1.30). Durante este paso, los parámetros de las funciones de membresía permanecen fijos.
2. Propagar nuevamente todos los patrones y actualizar los parámetros de las funciones de pertenencia mediante retropropagación usando (1.36)-(1.37). Durante este paso, los coeficientes de los consecuentes permanecen fijos.
3. Si el error se reduce en cuatro pasos consecutivos, incrementar la tasa de aprendizaje en un 10%. Si el error es sometido a combinaciones consecutivas de incremento y disminución, entonces reduzca la tasa de aprendizaje en un 10%.
4. Parar si el error es bastante pequeño, de lo contrario, continuar con el paso 1.

Para aplicaciones prácticas, el proceso de aprendizaje se incorpora en un proceso de diseño de tres etapas. Primero, un conjunto de datos de entrada-salida, para usarse como datos de entrenamiento, necesita ser generado u obtenido por el proceso. Otro conjunto opcional de datos puede usarse como prueba de datos después del entrenamiento para evaluar el desempeño del proceso de aprendizaje. Segundo, es necesario crear estructuras iniciales para el sistema difuso. Para cada entrada, el rango de operación número de funciones de membresía, al igual que su forma, debe definirse. Por último, el proceso de aprendizaje se lleva a cabo usando el conjunto de datos de entrenamiento para ajustar las funciones de membresía y para determinar los parámetros de los consecuentes.

El sistema difuso resultante se verifica usando el conjunto de datos de prueba. La totalidad de este procedimiento de aprendizaje se tiene disponible en la librería de sistemas difusos de Matlab/Simulink [Jang, 2000].

1.5. Estrategias para generación de patrones de entrenamiento

La obtención de los M patrones entrada y salida, que son utilizados como datos de entrenamiento para la red del sistema neurodifuso, se realiza en dos versiones esencialmente diferentes. En la primera versión los datos se obtienen a partir de la ecuación del mapeo estático implícitamente definido por el controlador PI convencional. En la segunda versión, los datos se obtienen por medición directa en las entradas y salidas del controlador PI convencional. Los patrones de entrada y salida para la verificación del sistema neurodifuso son obtenidos de la misma manera.

1.6. Obtención de datos a partir de mapeo estático

Los datos de entrenamiento se obtienen por medio de la ecuación (1.10), que establece el mapeo estático, $R^2 \rightarrow R$, definido por el controlador PI convencional. Primeramente, los patrones de entrada se definen como puntos en el plano $e \times \Delta e$ en un rectángulo definido por los intervalos $e_{\min} \leq e \leq e_{\max}$ y $\Delta e_{\min} \leq \Delta e \leq \Delta e_{\max}$. Los puntos se distribuyen uniformemente en el área del rectángulo, para lo cual se considera un número determinado (p. ej. 10) de puntos en cada intervalo formando los conjuntos de puntos:

$$E = \{e_{\min}, e_{\min} + \delta e, e_{\min} + 2\delta e, \dots, e_{\min} + n\delta e = e_{\max}\} \quad (1.38)$$

$$\Delta E = \{\Delta e_{\min}, \Delta e_{\min} + \delta \Delta e, \Delta e_{\min} + 2\delta \Delta e, \dots, \Delta e_{\min} + n\delta \Delta e = \Delta e_{\max}\} \quad (1.39)$$

$$\delta e = \frac{e_{\max} - e_{\min}}{n}, \quad n_{\text{entero}}$$

$$\delta \Delta e = \frac{\Delta e_{\max} - \Delta e_{\min}}{n}, \quad m_{\text{entero}} \quad (1.40)$$

Así, los patrones de entrada son los puntos definidos por el producto cartesiano de los conjuntos E y Δ :

$$E \times \Delta E = \{(x, y) | x \in E\} \quad (1.41)$$

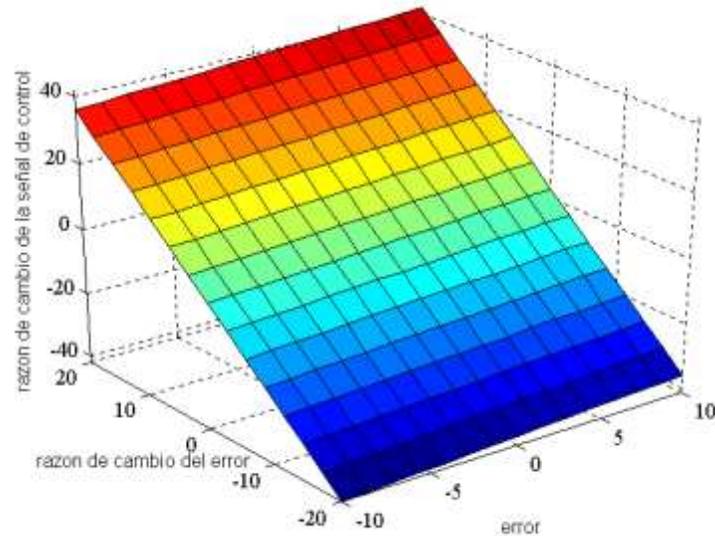
Entonces, los patrones de salida se obtienen calculando Δu para cada punto $E \times \Delta E$ usando la ecuación del mapeo estático (1.10).

Por lo tanto, el conjunto de patrones de entrada y salida es el conjunto

$$\{(e, \Delta e; \Delta u)\} = \{(x, y, z) | (x, y) \in E \times \Delta E, z = K_i x + K_p y\} \quad (1.42)$$

Como un ejemplo, supóngase que $K_1 = .1$, $K_p = 3.5$, $e_{\max} = 10$, $e_{\min} = -e_{\max} = -10$, $\Delta e_{\max} = 20$, $\Delta e_{\min} = -\Delta e_{\max} = -20$, $n = 20$, $m = 40$. Con estos datos el conjunto de patrones de entrada y salida determina el plano en R^3 que se muestra en la Figura 1.8. Del mismo modo, el sistema neurodifuso que se obtenga con este conjunto de datos reproducirá el mismo plano (Figura 4.8)

Figura 4.8 Mapeo estático definido por el controlador PI convencional

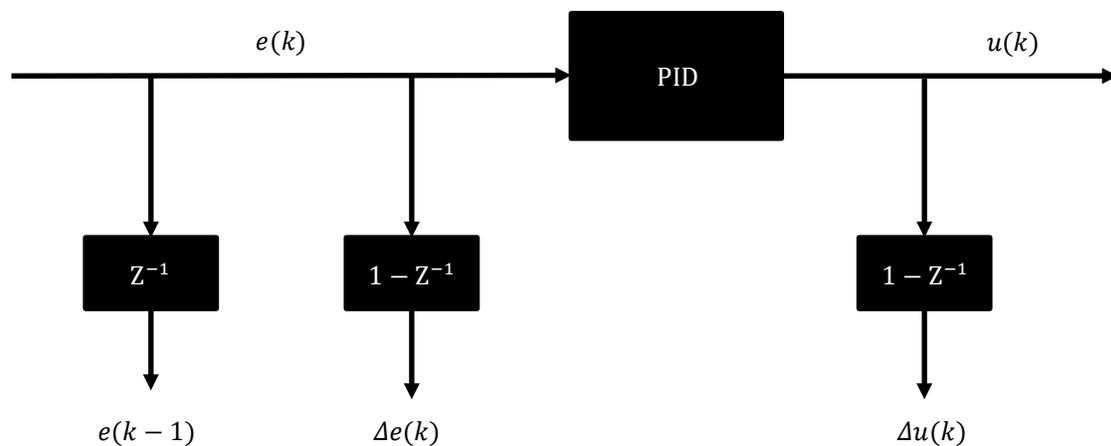


Aunque muy conveniente, esta forma de generar los patrones de entrenamiento requiere del conocimiento preciso de los parámetros del controlador PI convencional y que la estructura de ambos controladores PI, convencional y neurodifuso, sea la misma. Estos requerimientos podrían limitar la aplicación de este método en un caso real.

1.7. Obtención de datos por medición directa

Esta versión para la obtención de los datos de entrenamiento se desarrolló para evitar las limitaciones de la versión anterior y darle mayor versatilidad al método de diseño de controladores PI neurodifusos que la usen. Los patrones de entrada y salida se obtienen por medición directa en la(s) entrada(s) y la salida del controlador PI convencional, tal y como se muestra en la Figura 4.9

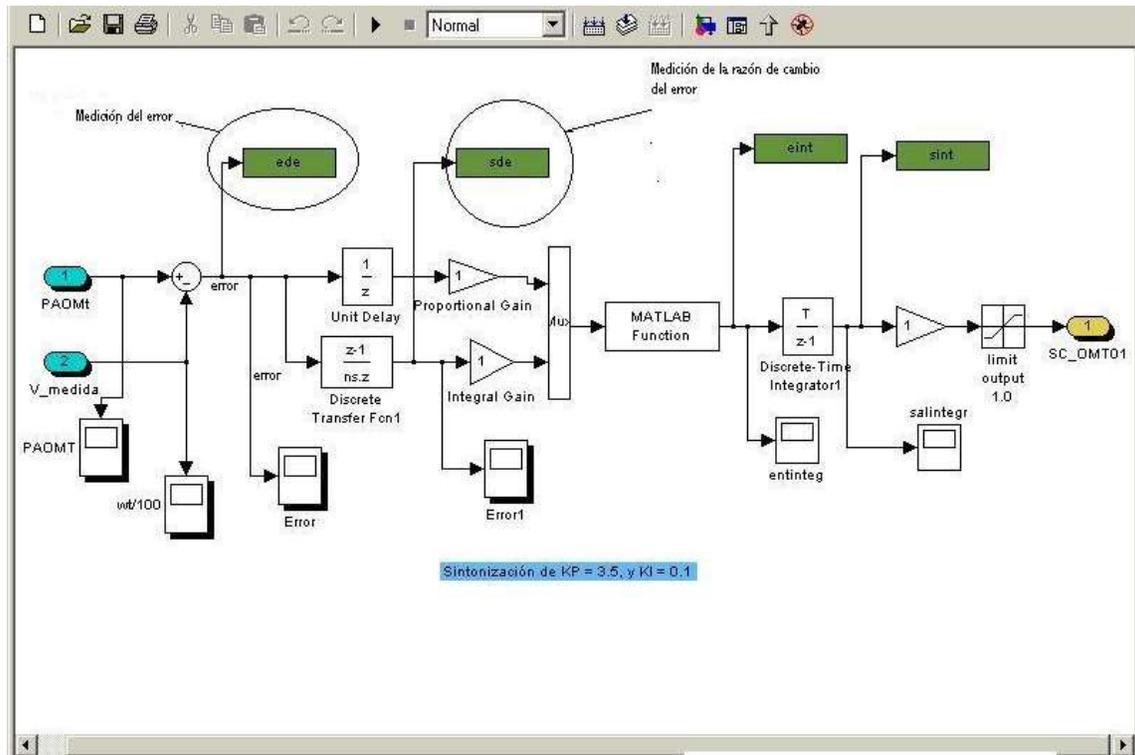
Figura 4.9 Medición directa de datos entrada-salida



Posiblemente, la única limitación de esta versión sería la dificultad del acceso a las señales del error y de control. Se obtuvieron los patrones de datos de entrada y salida para el desarrollo del controlador PI neurodifuso para la UTG. La medición de los datos de entrada y salida se llevó a cabo directamente, en el ambiente de simulación de la UTG en Simulink. El diagrama de simulación está compuesto por tres secciones principales: control, secuenciador y turbogenerador. Las variables de entrada y salida deseadas se encuentran localizadas en la sección de control, por lo que en esa sección se realizaron los arreglos correspondientes para la lectura de datos entrada-salida (Figura 4.10).

Los datos de entrada y salida se recolectan durante la simulación del arranque de la UTG, asegurándose que los datos de entrada y los de salida sean recolectados en los mismos instantes de muestreo. Los parámetros recolectados son almacenados en un archivo para su utilización posterior en el diseño del sistema neurodifuso del controlador.

Figura 4.10 Diagrama de simulación de la UTG en Simulink



1.8. Realización de sistemas de inferencia

Una vez que se cuenta con los datos de entrenamiento, obtenidos en la Sección 1.5, se puede proceder a crear los sistemas de inferencia. Los datos de entrenamiento generados a partir del mapeo estático son utilizados para crear dos sistemas neurodifusos que son utilizados para el control de un sistema de primer orden con retardo. Por otra parte, los datos de entrenamiento obtenidos por medición directa son utilizados para construir otros dos sistemas neurodifusos para el control de una UTG. En todos los casos, los sistemas difusos son del tipo TSK, con fusificación tipo singletón, inferencia basada en reglas individuales, implicación por operador mínimo, norma-t: operador mínimo, norma-s: operador máximo, y composición de la salida por promedio ponderado. Las diferencias consisten en el tipo de consecuentes: lineal o constante. Las diferencias consisten en el tipo de consecuentes: lineal o constante. Los cuatro sistemas resultantes se describen a continuación.

Sistemas difusos de datos por mapeo.

Para el caso de reglas con consecuentes lineales, el sistema difuso TSK obtenido con $K_p=3.5$ y $K_i=1.1$. Las funciones de pertenencia del error y el cambio del error se muestran en las figuras 4.11 y 4.12, respectivamente.

Figura 4.11 Valores lingüísticos del error

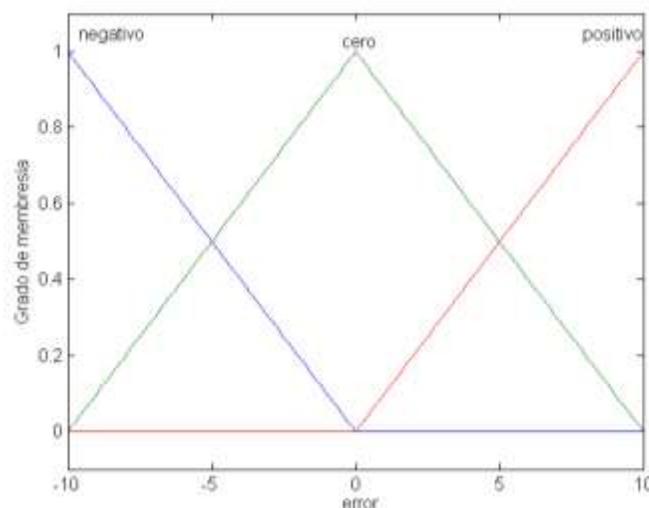
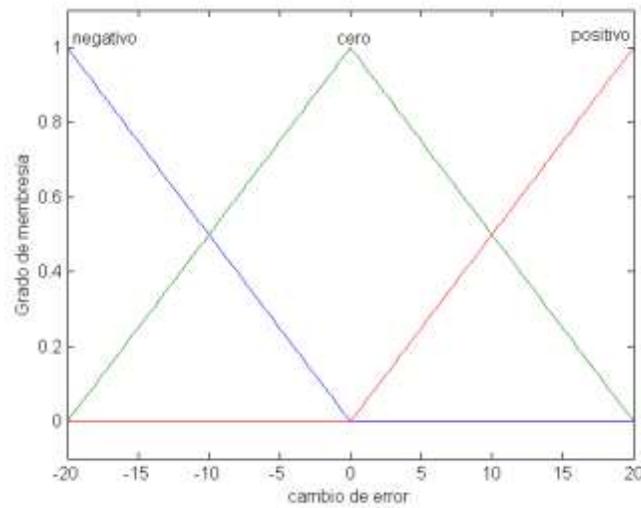


Figura 4.12 Valores lingüísticos de la razón del cambio del error

Los coeficientes de los consecuentes lineales se resumen en la Tabla 4.2, en donde los C_i , para $i= 1, 2, \dots, 9$ representan los coeficientes de la i -ésima fila de la Tabla 4.1

Tabla 4.1 Coeficientes de los consecuentes lineales (dados por mapeo)

No. de regla	P	q	K
1	4.09054210	6.49434637	-1.10065470
2	4.09054184	6.49434595	-.61358211
3	4.09054126	6.49434576	-.1265027
4	4.09054136	6.49434618	.48707679
5	4.09054198	6.49434625	2.10257901
6	4.09054146	6.49434626	.48707514
7	4.09054100	6.49434589	.12650775
8	4.09054200	6.49434602	.61358046
9	4.09054172	6.49434618	1.10065966

Tabla 4.2 Reglas del sistema neurodifuso (consecuentes lineales, dados por mapeo)

de/dt	e			
	Du	Negativo	Cero	Positivo
Positivo	C_1	C_2	C_3	C_4
Negativo	C_5	C_6	C_7	C_8
Cero	C_9	C_{10}	C_{11}	C_{12}

Para el caso del TSK con reglas con consecuentes constantes, las funciones de pertenencia del error y el cambio del error se muestran en las Figuras 4.13 y 4.14, respectivamente.

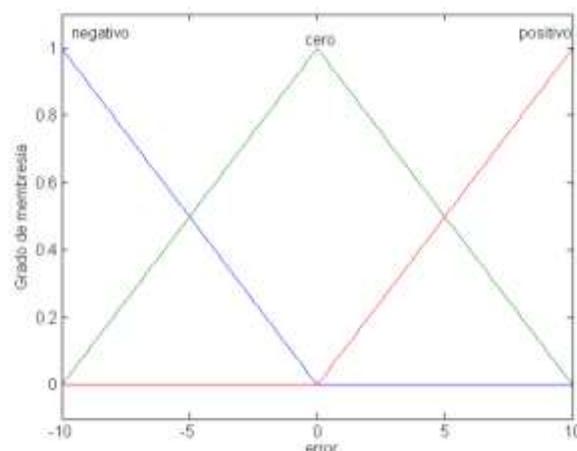
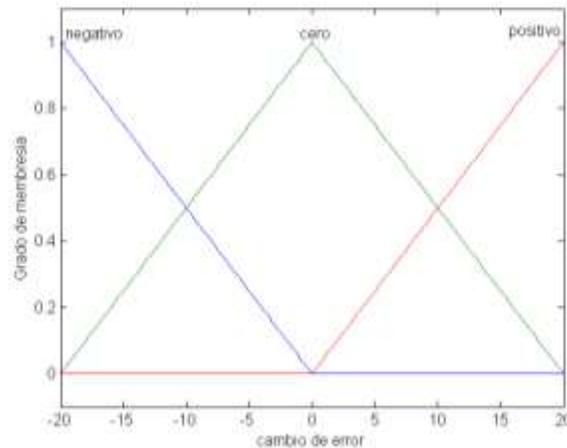
Figura 4.13 Valores lingüísticos del error

Figura 4.14 Valores lingüísticos de la razón del cambio del error**Tabla 4.3** Reglas del control neurodifuso

		e		
		Negativo	Cero	Positivo
de/dt	Positivo	41.71099477482	2.792999923536	36.12499534375
	Cero	38.91799934850	7.7979627e-008	38.91799966870
	Negativo	36.12499558346	2.793000243738	41.71099370242

		e		
		Negativo	Cero	Positivo
de/dt	Positivo	41.71099	2.792999	36.12499
	Cero	38.91799	7.797e-8	38.91799
	Negativo	36.12499	2.793000	41.71099

1.9. Agradecimiento

Este trabajo fue apoyado por el Instituto Nacional de Electricidad y Energías Limpias, SEPI-ESIME y el CONACYT.

1.10. Conclusiones

En este capítulo se presentó un controlador neurodifuso que emula la dinámica de un controlador PI convencional en configuración de velocidad. El sistema de inferencia difuso embebido en el controlador es del tipo TSK de primer orden e implementa un mapeo $R^2 \rightarrow R$, de la señal de error y su razón de cambio a la diferencial de la señal de control. El diseño del sistema de inferencia difuso se lleva a cabo fuera de línea empleando el método ANFIS con patrones de aprendizaje obtenidos de las entradas y la salida de un controlador PI convencional.

Los resultados obtenidos en las pruebas de análisis muestran que el controlador neurodifuso digital posee buenas características de desempeño, lo cual demuestra la factibilidad de emplearlo en el mismo tipo de aplicaciones que un controlador PI convencional mediante una sustitución directa.

Después de la demostración de factibilidad del controlador neurodifuso reportada, se tiene planeada su utilización para el control de la velocidad durante el arranque y el control de potencia generada de una unidad turbogás, en donde se tendrá en cuenta la disponibilidad de un gran número de parámetros para mejorar el rendimiento de la unidad bajo índices de desempeño específicos.

1.11. Referencias

Aström, K. J. y Hägglund, T. (1993). Automatic tuning an adaptation for PID controllers a survey. Control Engineering Practice. Vol. 1, No. 4, págs. 699-714.

- Azencott, R. (Editor) (1992). *Simulated Annealing: Parallelization Techniques*. John Wiley & Sons Inc. París, Francia.
- Berenguel, M. y Camacho, E.F. (1994). *Advances in Model Based Predictive Control. Application of generalized predictive control to a solar power plant*. Ed. D.W. Clarke. Oxford, Inglaterra.
- Berenji, H. y Khedkar, P. (1993). *Learning and tuning fuzzy logic controllers through reinforcements*. IEEE Transactions on Neural Networks. Vol. 3, No. 5, págs. 724-740.
- Carretero, M. y De Lara S. (2001). *Sistema para Desarrollo y Validación de Algoritmos de Control para Turbinas de Gas*. Memorias XIV. Reunión de Verano de Potencia, Aplicaciones Industriales y Exposición Industrial, RVP-AI/2001. Acapulco, México.
- Chávez, R. y Delgadillo, M.A. (1990). *Modernization of a computer control system in a combined cycle power plant*. Proceedings ISA 90. Nueva Orleans, E.E.U.U. Vol. 4, págs. 165-170.
- Chávez, R. y Díaz, R. (1993). *Upgrading of a computer control system in a combined cycle power plant*. Proceedings Power-Gen Americas '93. Dallas, E.E.U.U. Vol. 12-13, págs. 56-62.
- Chávez, R. y Gómez, O. (1995). *Sistemas de control distribuido en plantas de ciclo combinado*. Boletín IIE. Vol. 19, No. 2, págs. 47-51.
- Comisión Federal de Electricidad. (1990). *Manual de Operación de la Turbina de Gas*. Manual interno. Gómez Palacio, México.
- De Lara, S. y Martínez, M.A. (1999). *Desarrollo del sistema para la sintonización de controladores difusos usando redes neuronales artificiales*. IIE reporte interno. No. 11500INDT (A5)01.03. Cuernavaca, México.
- Delgadillo, M.A. y Hernández, M.A. (2002). *Modeling and Dynamic Simulation of a Gas*. Proceedings ISA 2002. San Francisco, E.E.U.U. Vol. 2, págs. 126-132.
- García, C.D. (1997). *Control Digital de Unidades Turbogás*. Tesis de Maestría. Centro Nacional de Investigación y Desarrollo Tecnológico. Cuernavaca, México.
- García, C.D. y Garduño R. (1997). *Gas turbine fuzzy speed control*. Proceedings 2nd Joint Mexico-USA International Conference on Neural Networks and Neurocontrol. Sianka'an, México.
- Garduño, R. y Sánchez, M. (1995). *Control system modernization: turbogás unit case study*. Symposium on Control of Power Plants and Power Systems. Cancún, México. Vol. 2, págs. 245-250.
- Giacomini, L. Bordbar, B. y Holding, D.J. (2000). *Hybrid sliding mode control for a set of interconnected nonlinear*. Proceedings 39th IEEE Conference on Decision and Control. Sydney Australia. Vol. 3, págs. 2180-2185.
- Glover, F. (1986). *Future paths for integer programming and links to artificial intelligence*, *Computers and Operations Research*. Vol.5, págs. 533-549.
- Goldberg, D.E. (1989). *Genetic algorithms in search, optimization machine learning*. Adison-Wesley, Publishing Co. Reading. E.E.U.U.
- Guo, L. (1994). *Further results on least squares based adaptive minimum variance control*. SIAM Journal on Control and Optimization, Society for Industrial and Applied Mathematics. Kansas, E.E.U.U. Vol. 32, No. 1, págs. 187-212.
- Halgamuge y Glesner, M. (1993). *The fuzzy neural controller FuNe II with a new adaptive defuzzification strategy based on CBAD distributions*. In European Congress on Fuzzy and Intelligent Technolodcies'93. Aachen, Alemania. Vol. 2, págs. 852-867.

- Jang, R. y Shing, J. (1993). *ANFIS: Adaptive Network Based Fuzzy Inference System*. IEEE Transactions on Systems Man and Cybernetics. Vol. 23, No. 3, págs. 665-685.
- Jang R. (2000). *Fuzzy Logic Toolbox*. Ed. Natick,MS, Mathworks. E.E.U.U.
- Kennedy, J. y Eberhart, R.C. (1995). *Particle swarm optimization*. Proceeding IEEE International Conference on Neuronal Networks. Perth, Australia. Vol. 4, págs. 1942-1948.
- Lee, J. F. (1992). *Theory and design of Steam and Gas Turbines*. Mc- Graw-Hill Inc. New York, E.E.U.U.
- Naucke, D. y Klawonn, F. (1997). *Foundations of Neurofuzzy Systems*. John Wiley & Sons Inc. París, Francia.
- Nürnbergger, A. Nauck, D. y Kruse, R. (1999). *Neuro-fuzzy control based on the NEFCON model recent developments*. Soft Computing 2. Vol.4, págs. 168-182.
- Pat, M. y de Lara, S. (1999). *Sintonización con redes neuronales de un controlador PID*. Aplicación a una unidad tubogás. Información Tecnológica. Vol. 10, No. 6, págs. 25-32.
- Ramírez, G.M. (1996). *Modernización del Control de Velocidad, Potencia y Temperatura de una Unidad Turbogás*. Tesis de Licenciatura. Instituto Politécnico Nacional. D.F, México.
- Rugh, W. J. (1991). *Analytical framework for gain scheduling*. IEEE Control Systems Magazine. Vol. 11, No. 1, págs. 79-84.
- Sánchez, M. y Bahamaca L. (1999). *Realización de controladores lógicos difusos para el control de procesos en tiempo real: especificación y diseño*. Memorias 2o. Encuentro Nacional de Computación. Pachuca, México.
- Sánchez, M. y Bahamaca L. (2000). *Speed and load fuzzy control for gas turbines*. Proceedings 1 st. ISA POWID/EPRI/DOE Joint Controls and Instrumentation Conference. Snt. Antonio U.S.A. Vol. 1, págs. 2180-2185.
- Sánchez Reinoso, M. A. (2019). *Comparación del rendimiento entre los reguladores P+ I incremental, Fuzzy Mamdani Incremental y control de modelo interno para el mando de una planta Quanser*. (Master's thesis, Escuela Superior Politécnica de Chimborazo).
- Sugeno, M. (1985). *An introduction survey of fuzzy control*. Inform. Sci., Vol. 36, No. 1, págs. 59-83.
- Takagi H. and Hayashi I. (1991), *NN-driven fuzzy reasoning*. Int. J. Approximate Reasoning Vol.5, No.3, págs.191-212.
- Trejo, C.G. (2000). *Sistema de Control Digital de una Unidad Turbogás Modelo GE- 5001*. Tesis de Licenciatura. Universidad del Sol. Cuernavaca, México.
- Yarizhkin, V. (1979) *Centrales Termoeléctricas*. Tomo 1 y 2 Ed. Nir Moscú. Moscú, Rusia.
- Wang, L.X. (1997). *A course in fuzzy systems and control*. Ed.Prentice Hall. E.E.U.U.