

DISEÑO DE UN CONTROLADOR NEURO DIFUZO PARA CONTROLAR LA VELOCIDAD DE UN MOTOR DE CORRIENTE CONTINUA

Juan Sebastian Toro Monsalve

Universidad Tecnológica de Pereira

Facultad de Ingenierías

Programa de ingeniería Eléctrica

Pereira

2015

**DISEÑO DE UN CONTROLADOR NEURO DIFUZO PARA CONTROLAR LA
VELOCIDAD DE UN MOTOR DE CORRIENTE CONTINUA**

Juan Sebastian Toro Monsalve

Proyecto de grado presentado para optar por el título de Ingeniero Electricista

Director

MEE. Didier Giraldo Buitrago

Universidad Tecnológica de Pereira

Facultad de Ingenierías

Programa de ingeniería Eléctrica

Pereira

2015

DEDICADO

A mi madre Marleny Monsalve Ramirez, mi padre Wilson Toro Lopez por todo el apoyo brindado durante mi desarrollo profesional

Juan Sebastian

AGRADECIMIENTOS

Al ingeniero Didier Giraldo y Johanna A. Chavez Toro por su acompañamiento en el desarrollo del proyecto

Juan Sebastian

Índice general

Índice General	I
----------------	---

Índice de Figuras	I
-------------------	---

REDES NEURONALES	36
1.1. Introducción	36
1.2. Función de Transferencia.....	38
1.3. Topologías de la Redes Neuronales Artificiales	39
1.3.1. Red de una Capa	40
1.3.2. Redes Multicapa.....	40
1.4. APRENDIZAJE.....	42
1.4.1. APRENDIZAJE SUPERVISADO	42
1.4.1.1. Perceptrón	43
1.4.1.2. Adaline.....	43
1.4.1.3. “Backpropagation”	43
1.4.2. APRENDIZAJE NO SUPERVISADO	44
1.4.2.2. Mapas Auto-Organizativos de Kohonen.....	45
1.4.2.3. Learning Vector Quantization	45
LÓGICA DIFUSA.....	47
2.1. Introducción.....	47
2.2. CONJUNTOS DIFUSOS	48
2.3. Función de membresía.....	49
2.4. Variable lingüística.....	50
2.5. Reglas base	51
2.6. Operaciones básicas entre conjuntos difusos	52
2.6.2. Complemento	52
2.6.2. La unión de dos conjuntos difusos.	52

2.6.3. La intersección de dos conjuntos difusos	53
2.7. Sistemas de control basados en lógica difusa.....	53
2.7.1. Base de conocimiento.....	55
2.7.2. Interfaz de difusión (Fuzificación)	57
2.7.3. Interfaz de concreción (Defuzificación)	60
SISTEMAS NEURO DIFUSO.....	63
3.1. Introducción.....	63
3.2. MÉTODO DEL GRADIENTE	64
3.2.1. MÉTODO DEL GRADIENTE CONJUGADO.....	66
3.3. ESTRUCTURA DEL SISTEMA NEURO-DIFUSO	67
3.4. ALGORITMO DE APRENDIZAJE USANDO GRADIENTE DESCENDIENTE (Backpropagation)	68
3.5. DISEÑO DEL CONTROLADOR NEURO-DIFUSO	70
3.5.1. CONDICIONES INICIALES	70
3.5.2. Base de datos	71
3.6. Implementación del sistema de control Neuro Difuso	73
3.6.1. Comportamiento del sistema de control	73
3.6.2. Caso sobreamortiguado	73
3.6.3. Caso amortiguado	76
3.6.4. Caso subamortiguado	78
3.6.5. Caso oscilatorio	81
CONCLUSIONES.....	84
BIBLIOGRAFIA	86

Índice de figuras

Figura 1.1 Red neuronal biológica	37
Figura 1.2 Neurona artificial	37
Figura 1.3 Función de transferencia	39
Figura 1.4 Red Neuronal de una Capa CON S Neuronas.....	40
Figura 1.5 Redes Neuronales de Tres Capas	41
Figura 1.6 Diagrama Abreviado Red Neuronal de Tres Capas	42
Figura 2.1 Conjunto de días que pertenecen al fin de semana	48
Figura 2.2 Representación del conjunto de personas de Estatura Alta.....	49
Figura 2.3 Funciones de membresía comúnmente utilizadas.	50
Figura 2.4 Valores que puede tomar la variable <i>Velocidad</i> en un motor eléctrico.....	51
Figura 2.5 complemento de un conjunto difuso	52
Figura 2.6 Unión entre conjuntos difusos.....	53
Figura 2.7 Intersección entre conjuntos difusos	53
Figura 2.8 Sistema de control difuso en lazo cerrado.	54
Figura 2.9 Configuración del sistema difuso.....	55
Figura 2.10 Construcción de una función de membresía.	56
Figura 2.11 Resolución obtenida según el número de funciones de membresía.....	56
Figura 2.12 Base de reglas difusas.	57
Figura 2.13 Proceso de difusión de una medida de temperatura.	58
Figura 2.14 Representación gráfica de la evaluación de Regla.....	59
Figura 2.15 Lógica de decisión.	60
Figura 2.16 Concreción utilizando método del centro de gravedad.....	61
Figura 3.1.a Representación de la variable del error con los conjuntos difusos μ_{ei}	72
Figura 3.2 Diagrama de bloques del sistema Neuro difuso. Siendo $nerr$ la diferencia entre la señal de control deseada (uT) y la señal de control actual (u).....	73
Figura 3.3 Inicio de la prueba para el caso sobreamortiguado. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa.....	74
Figura 3.4 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo del cual la señal de salida alcanza la señal de referencia	74
Figura 3.5 Comportamiento de la velocidad del motor bajo una perturbación	75
Figura 3.6 Inicio de la prueba para el caso amortiguado. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa.	76
Figura 3.7 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo en el cual la señal de salida alcanza la señal de referencia.....	77
Figura 3.8 Comportamiento de la velocidad del motor bajo una perturbación	78
Figura 3.9 Inicio de la prueba para el caso subamortiguado. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa.	79

Figura 3.10 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo del cual la señal de salida alcanza la señal de referencia	79
Figura 3.11 Comportamiento de la velocidad del motor bajo una perturbación	80
Figura 3.12 Inicio de la prueba para el caso oscilatorio. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa.	81
Figura 3.13 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo del cual la señal de salida alcanza la señal de referencia	82
Figura 3.14 Comportamiento de la velocidad del motor bajo una perturbación	83

REDES NEURONALES

1.1. Introducción

Las Redes Neuronales Artificiales (RNA) son aquellas que emulan el comportamiento de las redes neuronales biológicas y se han estado utilizando desde hace más de 30 años para aprender estrategias de solución basadas en ejemplos de comportamiento típico de patrones. Estos sistemas no requieren que la tarea a ejecutar se programe, ellos generalizan y aprenden de la experiencia.

Las redes neuronales biológicas están constituidas por tres partes principales: dendritas, cuerpo de la célula o soma y axón. Las dendritas son fibras nerviosas que se encargan de la recepción de señales eléctricas y llevan dichas señales al soma, este último cumple la función de acumular estas señales de entrada y las transfiere al axón el cual las entrega a las neuronas adyacentes. El punto de contacto entre el axón de una neurona y una dendrita de otra es conocido como sinapsis. Ver Figura 1.1

A partir de las características de una red neuronal biológica se plantea el modelo equivalente para la red artificial, así una RNA está constituida principalmente por un vector de entradas P_i que representa las señales que provienen de otras neuronas y que son capturadas por las dendritas, una matriz de pesos W_i que representan la intensidad de la sinapsis que conecta dos neuronas y una función suma que representa la función umbral que la neurona debe sobrepasar para activarse, éste último proceso ocurre en el cuerpo de la célula.

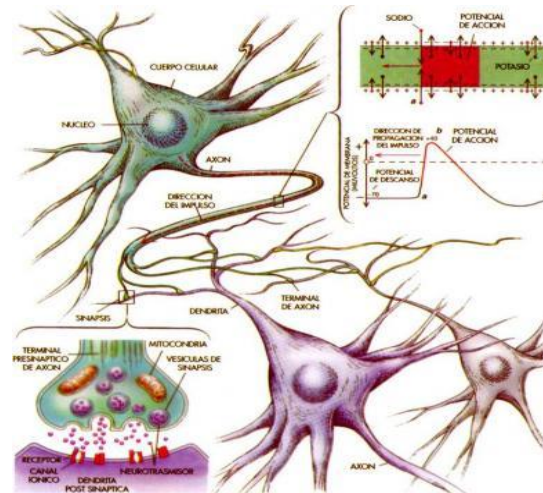


Figura 1.1 Red neuronal biológica

La Figura 1.2 muestra la estructura de una RNA de una neurona con múltiples entradas.

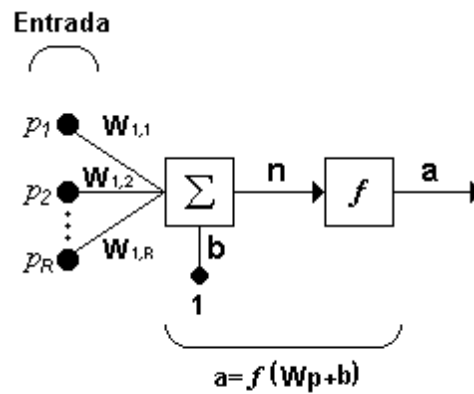


Figura 1.2 Neurona artificial

Donde b es una ganancia escalar que refuerza la salida del sumador n , este último representa la salida neta de la red neuronal, la salida total a está determinada por la función de transferencia f la cual puede ser una función lineal o no lineal de n y es escogida dependiendo de las especificaciones del problema que la neurona tenga que resolver. [1], [2]

La salida del sumador n se muestra en la ecuación (1.1).

$$n = w_{1,1} * p_1 + w_{1,2} * p_2 + \dots + w_{1,R} * p_R + b \quad (1.1)$$

Como se puede observar en (1.1), se adopta una convención en los índices de los elementos de la matriz de pesos, así el primer índice indica la neurona a la cual va dirigido el peso w y el segundo índice indica la fuente de la señal que alimenta la neurona, es decir, los índices $w_{1;2}$ representan la conexión de ese peso con la primera neurona desde la segunda fuente.

Escribiendo la ecuación (1.1) de forma matricial, se obtiene (1.2).

$$n = W_p + b \quad (1.2)$$

En este caso solo se tiene una neurona, por lo tanto la matriz W tiene una sola fila y la salida de dicha neurona está definida por (1.3). [2]

$$a = f(W_p + b) \quad (1.3)$$

1.2. Función de Transferencia

En el caso específico de las Redes Neuronales Artificiales existen muchas funciones de transferencia que convierten el estado de activación de la neurona en una señal de salida. Estas pueden ser o no de tipo lineal y su escogencia depende de la necesidad de la planta y de los criterios o parámetros de diseño. En el desarrollo de esta sección se hará una descripción de las clases de funciones de transferencia mencionadas anteriormente y que son de común aplicación en las RNA, debido a que presentan una respuesta satisfactoria ante las posibles señales de entrada.

En la Figura 1.3 se muestran las principales funciones de transferencia que son utilizadas en las Redes Neuronales Artificiales. [2]










Nombre	Relación Entrada/Salida	Icono	Función en Matlab
Limitador Fuerte	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Limitador Fuerte Simétrico	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Lineal	$a = n$		purelin
Lineal Saturado	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Lineal Saturado Simétrico	$a = -1 \quad n < 0$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Sigmoidal Logarítmico	$a = \frac{1}{1 + e^{-n}}$		logsig
Tangente Sigmoidal Hiperbólica	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Lineal Positiva	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitiva	$a = 1$ neurona con n máx $a = 0$ el resto de las neuronas		compet

Figura 1.3 Función de transferencia

1.3. Topologías de la Redes Neuronales Artificiales

Comúnmente una neurona está conformada por múltiples entradas y en ocasiones esto no es suficiente para resolver un problema, es por esta razón que se pueden necesitar varias neuronas operando en paralelo. Este arreglo es conocido como capa y existen redes neuronales constituidas por una, dos o más capas, estas últimas son denominadas redes multicapa.

1.3.1. Red de una Capa

Una red de una capa con S neuronas se muestra en la Figura 1.4 y como se puede observar cada elemento del vector de entrada \mathbf{p} está conectado a cada neurona por medio de la matriz de pesos W la cual está conformada por S filas. Además se tiene un vector de ganancias \mathbf{b} . [2]

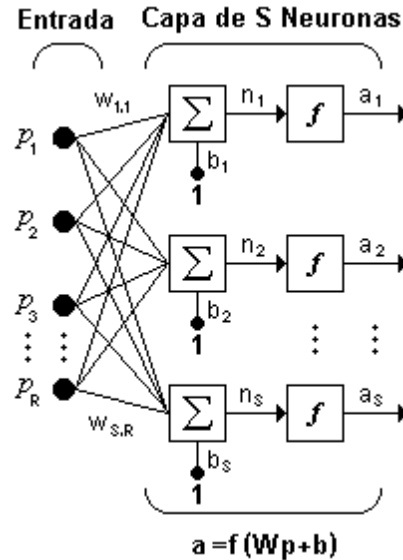


Figura 1.4 Red Neuronal de una Capa CON S Neuronas

Las dimensiones del vector de entrada \mathbf{p} , la matriz de pesos W y el vector de ganancias \mathbf{b} se presentan en (1.4).

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & \vdots \\ \cdots & \cdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad (1.4)$$

Cada neurona tiene una ganancia \mathbf{b}_i , un sumador, una función de transferencia f y una salida \mathbf{a}_i . Juntas estas salidas conforman un único vector de salida \mathbf{a} de la forma (1.5).

$$\mathbf{a} = f(W\mathbf{p} + \mathbf{b}) \quad (1.5)$$

1.3.2. Redes Multicapa

Ahora se considera una red con múltiples capas. Cada capa tiene su propio vector de entrada \mathbf{p} , su propia matriz de pesos W , su propio vector de ganancias \mathbf{b} y su propio vector de salida \mathbf{a} . [2]

Para esta configuración se adiciona la notación de superíndice para identificar las variables de cada capa, es decir, b_2^1 representa la ganancia de la segunda neurona de la primera capa. Así para todos los elementos de cada capa tal como se muestra en la Figura 1.5.

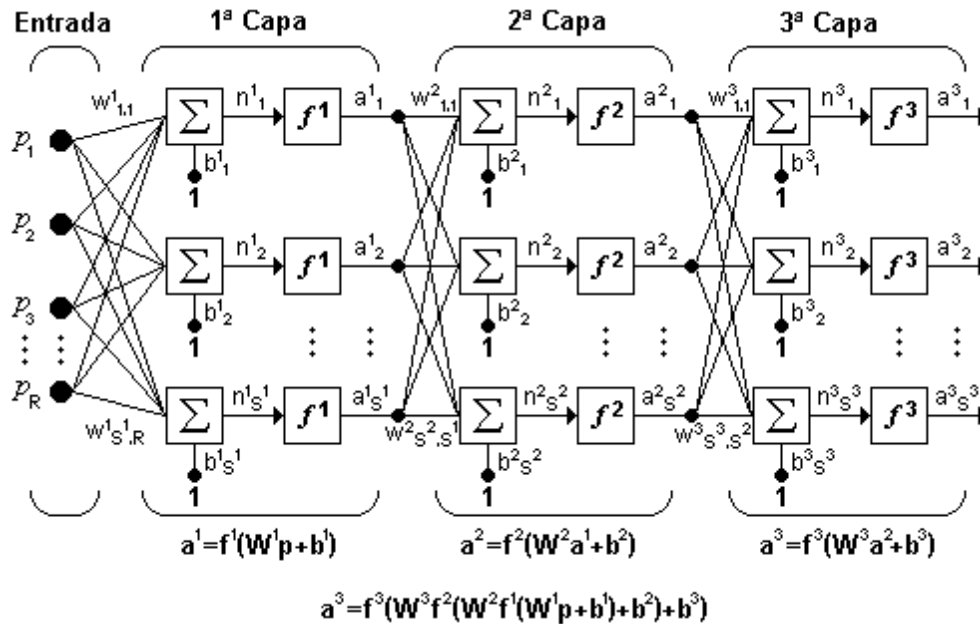


Figura 1.5 Redes Neuronales de Tres Capas

Es de notar que las salidas de la primera y segunda capa, son las entradas de la segunda y tercera capa, respectivamente. Además, cabe aclarar que cada capa puede tener una función de transferencia similar o diferente y así mismo, estar conformada por un número diferente de neuronas.

La salida total de la red es determinada por el conjunto de salidas de la capa 3 (ecuación (1.6)), la cual es denominada capa de salida y las capas 1 y 2 son denominadas capas ocultas.[2]

$$a^3 = f^3(w^3 f^2(w^2 f^1(w^1 p + b_1) + b_2) + b_3) \quad (1.6)$$

Para esta configuración se vuelve complejo mostrar todas las conexiones debido a que como ya se mencionó anteriormente, las capas pueden tener un número diferente de neuronas, es por esto que se acostumbra a tener un diagrama abreviado como se puede observar en la Figura 1.6. [2]

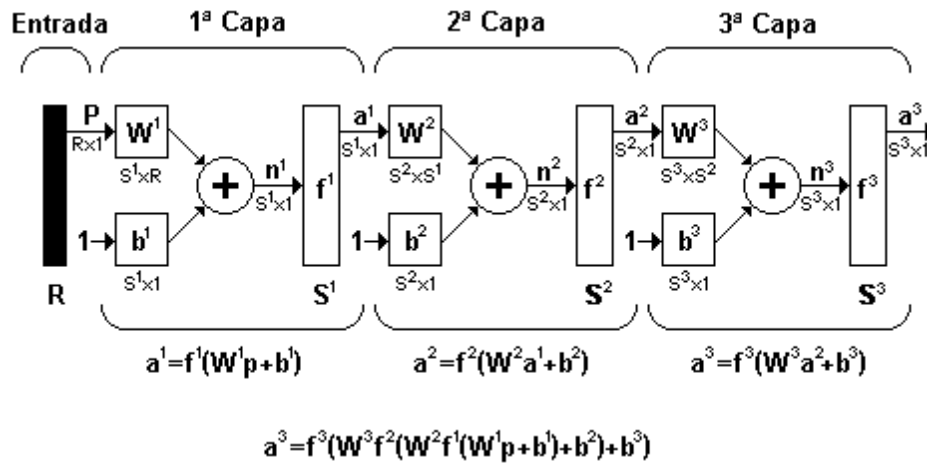


Figura 1.6 Diagrama Abreviado Red Neuronal de Tres Capas

Las redes de una capa tienen la limitante de solo resolver problemas linealmente separables, este término hace referencia a que la salida puede estar clasificada en dos valores diferentes, por esta razón surgieron las redes multicapa para superar dicha dificultad. [1]

1.4. APRENDIZAJE

Desde el punto de vista biológico, el aprendizaje es el proceso en el que se adaptan las sinapsis para que la red responda de un modo distinto a los estímulos del medio. En el contexto de RNA, el aprendizaje es un proceso mediante el cual la red modifica sus pesos en respuesta a una información de entrada. Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por lo tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables [14]. Un aspecto importante respecto al aprendizaje en las redes neuronales es el conocer cómo se modifican los valores de los pesos, es decir; cuál es la regla de aprendizaje para cambiar el valor asignado a las conexiones (pesos) cuando se pretende que la red aprenda una nueva información [14]. Por lo tanto una clasificación de las RNA se realiza de acuerdo al tipo de aprendizaje utilizado. De esta manera se tienen: redes neuronales con aprendizaje supervisado y no supervisado. [15]

1.4.1. APRENDIZAJE SUPERVISADO

En el aprendizaje supervisado, se presenta una entrada a la capa de entrada junto con una respuesta deseada para la capa de salida, el proceso de aprendizaje se basa en la comparación entre la salida que calculó la red y la respuesta deseada, generando un error, el error se utiliza para cambiar los parámetros de la red (pesos) de modo que resulte un mejor rendimiento. [15]

A continuación se darán algunos ejemplos de redes con aprendizaje supervisado.

1.4.1.1. Perceptrón

La red tipo Perceptron es un modelo matemático de una neurona biológica propuesto por Frank Rosenblatt en 1959 en su obra *Principles of Neurodynamics*, los Perceptrones son redes de propagación hacia adelante basados en unidades binarias. El objetivo de esta red es aprender, usando parte de la información, de tal modo que siempre que se presente un patrón de entrada p la salida sea la salida esperada, digamos t . [15]

1.4.1.2. Adaline

Al mismo tiempo que Frank Rosenblatt trabajaba en el modelo del Perceptron, Bernard Widrow y su estudiante Marcian Hoff introdujeron el modelo de la red Adaline y su regla de aprendizaje llamada algoritmo LMS (Least Mean Square). La red ADALINE es similar al Perceptron, excepto en su función de transferencia, la cual es una función de tipo lineal en lugar de un limitador fuerte como en el caso del Perceptron. Esta red presenta la misma limitación del Perceptron, ambas redes sólo pueden resolver problemas linealmente separables, sin embargo, el algoritmo LMS es más potente que la regla de aprendizaje del Perceptron ya que minimiza el error cuadrático medio, la regla sirvió de inspiración para el desarrollo de otros algoritmos, éste es el gran aporte de esta red. El significado del término ADALINE cambió ligeramente a finales de los años sesenta cuando decayó el estudio de las redes neuronales, inicialmente se llamaba ADaptive LInear NEuron (Neurona Lineal Adaptativa), para pasar después a ser ADaptive LInear Element (Elemento Lineal Adaptativo). La red ADALINE consta de una única neurona, como tal no es técnicamente una red neuronal, sino una neurona. La red ADALINE es adaptativa en el sentido de que existe un procedimiento bien definido para modificar los pesos con objeto de hacer posible que la red proporcione el valor de salida correcto para la entrada dada. [15]

1.4.1.3. “Backpropagation”

La red “Backpropagation” trabaja bajo aprendizaje supervisado, es decir, ésta necesita conocer la salida que se desea obtener. Una vez aplicado un estímulo a la entrada de la red, este se propaga capa por capa hasta generar una salida. La señal de salida obtenida se compara con la salida deseada calculando así una señal de error para cada componente de la salida total.

Una vez determinada dicha señal de error, ésta se propaga hacia atrás desde la salida, a través de todas las capas, hasta que todas las neuronas hayan recibido la señal de error que describe su contribución a la señal de error total. Basándose en la minimización del error medio cuadrático, el algoritmo “Backpropagation” actualiza los valores de las ganancias y de los pesos con el fin de que la red converja hacia un estado que le permita aprender los patrones de entrenamiento.

La importancia de este proceso consiste en que a medida que se entrena la red, las neuronas de las capas ocultas aprenden a reconocer diferentes propiedades del espacio total de entrada. Una vez finalizado el entrenamiento, aunque se presente un patrón arbitrario de entrada que contenga ruido o que la red no haya visto antes, dichas neuronas responderían con una salida activa si esta entrada contiene un patrón semejante a las características que aprendieron a reconocer durante su entrenamiento.

El algoritmo “Backpropagation” es comúnmente implementado en redes multicapa y como ésta se encuentre constituida no se determina por medio de alguna técnica particular, depende de la experiencia del diseñador y de las limitaciones computacionales que este posea. Para inicializar el proceso de entrenamiento se debe establecer la topología de la red, es decir, se debe definir el número de neuronas de la primera capa que depende directamente del número de componentes del vector de entrada, el número de capas ocultas y las neuronas que conforman cada una de ellas, el número de neuronas de la capa final la cual depende del número de componentes del vector de salida y la función de transferencia que precisa cada capa. Después de establecida la topología se asignan los valores iniciales de los parámetros de la red, tales como, el vector de entrada \mathbf{p} , la matriz de pesos \mathbf{W} y el vector de ganancias \mathbf{b} . [1], [15]

1.4.2. APRENDIZAJE NO SUPERVISADO

Las redes neuronales artificiales con aprendizaje no supervisado son aquellas que no necesitan un asesor externo para realizar su aprendizaje. La red no recibe información por parte del entorno que le indique si la salida generada en respuesta a una entrada es o no correcta. El aprendizaje no supervisado consiste en que la red descubra por si misma características, regularidades, correlaciones o categorías en los datos de entrada y se obtengan de forma codificada en la salida. En algunos casos, la salida representa el grado de similitud entre la información que se le está presentado en la entrada y la que se le ha mostrado en el pasado. En otro caso podría realizar un clustering o establecimiento de categorías, indicando con la salida de la red a qué categoría pertenece la información presentada como entrada, siendo la propia red quien deba encontrar las categorías apropiadas a partir de correlaciones en las entradas presentadas. [15]

A continuación se darán algunos ejemplos de redes con aprendizaje no supervisado.

1.4.2.1. *Red de Hopfield*

Una de las mayores contribuciones en el área de las redes neuronales fue desarrollada por el físico John Hopfield en la década de los 80, quien propuso un modelo neuronal no lineal, conocido como la red de Hopfield [15]. La red de Hopfield es presentada como un modelo de memoria asociativa de patrones, en el sentido de que es capaz de recuperar patrones almacenados a partir de información incompleta sobre los patrones o incluso a partir de patrones con ruido

1.4.2.2. Mapas Auto-Organizativos de Kohonen

Existen evidencias que demuestran que en el cerebro hay neuronas que se organizan en muchas zonas, de forma que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de mapas bidimensionales. Por ejemplo, en el sistema visual se han detectado mapas del espacio visual en zonas del córtex (capa externa del cerebro), también en el sistema auditivo se detecta una organización según la frecuencia a la que cada neurona alcanza mayor respuesta (organización tonotópica) [14]. Esto sugiere, por lo tanto, que el cerebro podría poseer la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior. A partir de estas ideas, Teuvo Kohonen presentó en 1982 un modelo de red neuronal con capacidad para formar mapas de características de manera similar a como ocurre en el cerebro. El objetivo de Kohonen era demostrar que un estímulo externo por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de los mapas; [14], [15]

El aprendizaje en el modelo de Kohonen es de tipo OFF-LINE, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento. Esta red utiliza un aprendizaje no supervisado de tipo competitivo, en el que las neuronas compiten por activarse y sólo una de ellas permanece activa ante una determinada información de entrada a la red. De esta manera los pesos de las conexiones se ajustan en función de la neurona que haya resultado vencedora. [15]

Durante la etapa de entrenamiento, se presenta a la red un conjunto de entrenamiento (vectores de entrenamiento) para que ésta establezca, en función de la semejanza entre los datos, las diferentes categorías (una por neurona), que servirán durante la fase de funcionamiento para realizar clasificaciones de nuevos datos que se presenten a la red. En el caso de existir más patrones de entrenamiento que neuronas, más de uno deberá asociarse con la misma neurona, es decir pertenecerán a la misma clase. En este modelo el aprendizaje no concluye después de presentarle una vez todos los patrones de entrada, sino que habrá que repetir el proceso varias veces para refinar el mapa topológico de salida, de tal forma que cuantas más veces se presenten los datos, tanto más se reducirán las zonas de neuronas que se deben activar ante entradas parecidas, consiguiendo que la red pueda realizar una clasificación más selectiva. [15]

1.4.2.3. Learning Vector Quantization

La red Learning Vector Quantization (LVQ), emplea tanto aprendizaje supervisado como no supervisado. Es una red bicapa en la que cada neurona de la primera capa es asignada a una clase, con varias neuronas generalmente asignadas a la misma clase. En la segunda capa cada clase es asignada a una neurona. El número de neuronas en la primera capa, S^1 , debe ser mayor o al menos igual que el número de neuronas en la segunda capa, S^2 [11], si se identifican N subclases, entonces, S^1 debe contener al menos N neuronas. Al igual que

con redes competitivas, cada neurona en la primera capa de la red LVQ aprende un vector prototipo, el cual permite a la neurona clasificar una región del espacio de entrada, sin embargo en lugar de calcular la distancia entre la entrada y el vector de pesos por medio del producto punto, la red LVQ calcula la distancia directamente. Una ventaja de hacer el cálculo de la distancia directamente, es que los vectores no necesitan ser normalizados; cuando los vectores son normalizados la respuesta de la red será la misma sin importar la técnica que se utilice. [15]

LÓGICA DIFUSA

2.1. Introducción

La lógica difusa surge gracias al profesor Lotfi Zadeh, matemático de la Universidad de California, Berkeley, en el momento en que publica un documento titulado Fuzzy Sets en la revista Information and Control [3]. Este documento estuvo terminado dos años antes de su publicación, pero no había sido divulgado debido a que esta revista técnica no se atrevía a publicar tan radical idea, pues era inconcebible en el campo de la ingeniería que para la época, se aferraba de la descripción matemática clásica de la lógica bivalente. La aceptación y publicación de este documento solo ocurrió hasta 1965, año en el que el profesor Zadeh asumió el cargo de editor de la revista y a partir del cual, propuso otras ideas sobre conjuntos y algoritmos difusos.

Esta teoría propuesta sobre conjuntos difusos, partió de observaciones realizadas por Zadeh para el modelado de sistemas y la limitante que en repetidas ocasiones se tenía para describir ciertos fenómenos mediante expresiones matemáticas, por tal motivo, plantea su principio de la incompatibilidad el cual predica que:

Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes.

De esta manera, Zadeh realiza este planteamiento teórico que denominó lógica difusa, definiéndolo como un intento de formalización o mecanización de dos importantes capacidades del ser humano. La primera, la capacidad de conversar, razonar y tomar decisiones racionales en un ambiente de imprecisión o información incompleta. Y la segunda, la capacidad de formular una amplia variedad de tareas físicas y mentales sin realizar mediciones y sin computadores [4].

La lógica difusa puede ser descrita como un sistema interpretativo, en el cual los objetos o elementos son relacionados con conjuntos de fronteras no nítidamente definidas, otorgándoles un grado de pertenencia relativa o graduada y no estricta como es costumbre en la lógica tradicional [5]. Se podría decir que existe una especie de interpolación entre una frontera y otra, o bien, entre un conjunto y otro, permitiendo así tener en cuenta elementos que presentan un estado ambiguo como el que se presenta en expresiones tales como: *La temperatura está más o menos alta o El vaso está medio lleno*, las cuales son evidentemente inaceptables en la lógica clásica o bivalente. Este tipo de situaciones son frecuentemente encontradas en la vida cotidiana como por ejemplo, en el caso de considerar los días que pertenecen al fin de semana como se puede ver en la figura 2.1. La mayoría estamos de acuerdo con que obviamente este conjunto incluye totalmente los días sábado y

domingo, pero ¿Que sucede con el día viernes? Muchos viven parte de este día como si fuera del fin de semana, pero técnicamente y razonando de la manera de la lógica clásica, este es excluido del fin de semana.

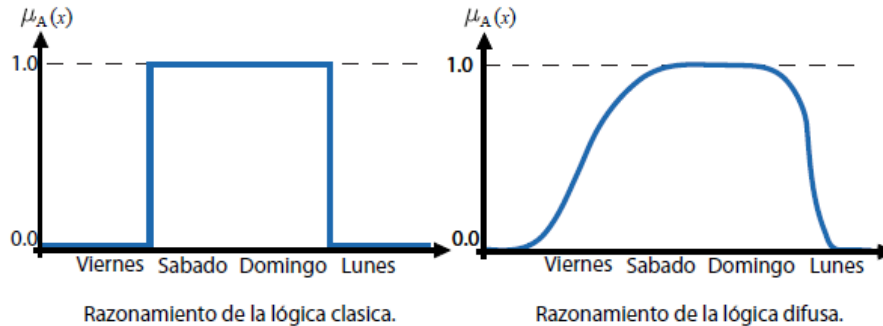


Figura 2.1 Conjunto de días que pertenecen al fin de semana

Se puede decir entonces, que la lógica difusa no es realmente difusa, de hecho, una de las principales contribuciones de ésta, es el alto poder de precisión y a su vez es mucho más cercana al razonamiento humano.

Existe una gran cantidad de planteamientos teóricos alrededor de la lógica difusa que han tenido una amplia aplicación en el diseño de sistemas de control, puesto que con esta herramienta se ha logrado incorporar sentencias del lenguaje común para interpretar el estado de las variables de cierto proceso, haciendo que el sistema de control pueda ser diseñado a partir de la descripción del comportamiento deseado por el operario. Para llevar a cabo este proceso, las entradas son clasificadas dentro de conjuntos difusos asignándoles en cada momento un grado de pertenencia relativa a estos conjuntos de fronteras no nítidamente definidas, después son sometidos a un conjunto de reglas para luego deducir en forma lógica una actuación posible que permita llevar la variable al estado deseado, produciendo una o varias acciones de salida (acción de control).

2.2. CONJUNTOS DIFUSOS

Un conjunto difuso es una clasificación de objetos con un grado de membresía continuo. Como conjunto, es caracterizado por una función de membresía (característica), la cual asigna a cada elemento u objeto un grado de membresía dentro del rango de cero a uno [0,1].

Sea $X = f\{x\}$ un conjunto clásico cuyos elementos son representados por x . Luego un conjunto difuso A en X representa un conjunto de pares ordenados:

$$A = (x, \mu_A(x)), x \in X$$

Un ejemplo pertinente para explicar el concepto de conjunto difuso, es el conjunto de valores que inciden en la expresión Esta persona es alta, haciendo referencia a la estatura de

una persona. Basándose en el razonamiento de la lógica clásica, el conjunto de estaturas para los cuales la anterior expresión es válida, para aquellas personas con estaturas mayores a un valor determinado como podría ser 1.70 metros.

Este concepto de conjuntos difusos, señala que todos los elementos en los que se basa el razonamiento humano son etiquetas cualitativas y no necesariamente cuantitativas las cuales pueden ser modeladas dentro de un lenguaje matemático a través de la teoría de conjuntos difusos. Para una expresión tal como Estatura Alta, la teoría de conjuntos difusos plantea que este conjunto no posee una frontera nítidamente establecida que discrimine de manera estricta si un valor de estatura pertenece a este conjunto o no, sino que de manera distinta, asigna a esta variable un grado de pertenencia a dicho conjunto, siendo de esta manera como una persona de 1.65 metros de estatura podría tener una pertenencia de 0.85 al conjunto de Estatura Alta y a la vez, un grado de pertenencia de 0.1 al conjunto de Estatura Baja como se puede ver en la figura 2.2.

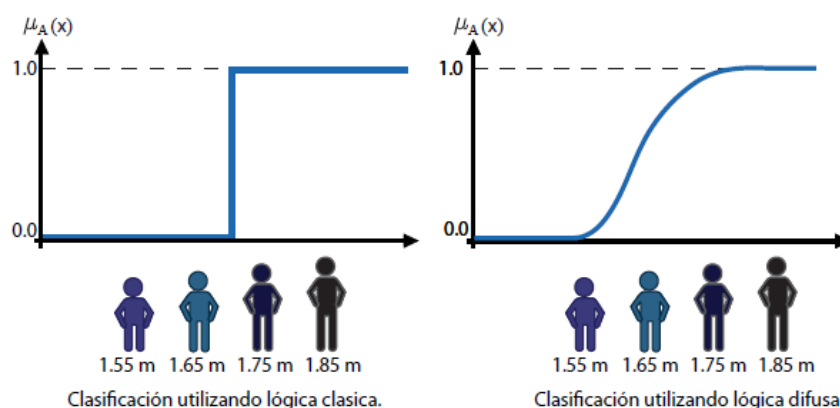


Figura 2.2 Representación del conjunto de personas de Estatura Alta.

Estos conjuntos no nítidamente definidos juegan un papel muy importante en el pensamiento humano, particularmente en los dominios de reconocimiento de patrones, comunicación de información y abstracción, permitiendo incorporar sentencias del lenguaje común para interpretar el estado de las variables de cierto proceso, asignándoles en cada momento un grado de pertenencia a estos conjuntos difusos.

2.3. Función de membresía

Las funciones de membresía constituyen un método utilizado para la representación de los conjuntos difusos, de tal manera que a cada punto en el espacio se le asigna un valor o grado de pertenencia a dicho conjunto, definiendo así al conjunto mismo. De este modo, $\mu_A(x)$ es el grado de membresía de x en A y es limitado a valores entre 0 y 1, siendo 0 el más bajo grado de membresía y 1 el más alto. Nótese el contraste con la teoría de conjuntos clásica o discreta, en el cual $\mu_A(x) = 0$ ó 1 , donde 0 significa que x no pertenece a A y 1 denota que x pertenece a A .

Las funciones de membresía más utilizadas son: triangular, trapezoidal, gaussiana y sigmoideal, como se muestra en la figura 2.3.

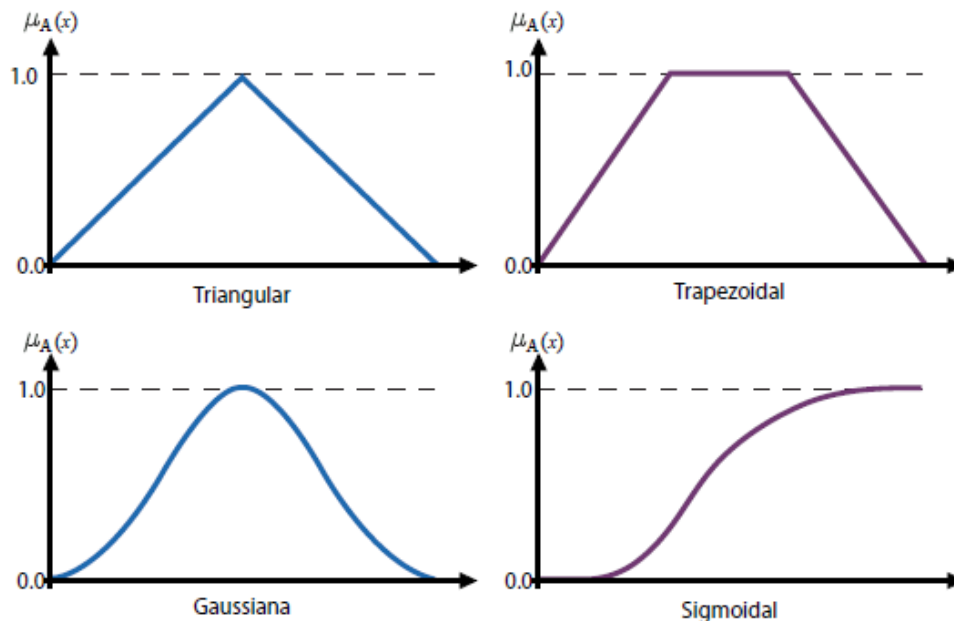


Figura 2.3 Funciones de membresía comúnmente utilizadas.

Existe cantidad de software especializado donde se podría encontrar gran variedad de funciones de membresía, pero la selección de una de éstas siempre estará determinada por el problema, análisis y punto de vista del diseñador y la capacidad de carga computacional, siendo este último un factor sumamente influyente en el momento de escoger la función de membresía, donde en la mayoría de los casos, la función triangular resulta siendo el candidato más llamativo.

2.4. Variable lingüística

Si una variable puede adoptar palabras en el lenguaje natural como sus valores, ella es llamada una variable lingüística, donde las palabras son caracterizadas mediante conjuntos difusos definidos en el universo de discurso en el cual la variable es definida [6].

Cuando se usan sensores para medir variables, ellos entregan información numérica para expresar su magnitud; mientras que en el caso de preguntar a una persona experta que evalúe la variable, ésta lo expresa en palabras. Es aquí donde toma gran importancia el concepto de variable lingüística, puesto que ésta permite realizar una descripción en lenguaje natural a través de términos matemáticos.

En el caso de la velocidad de un motor eléctrico, ésta es una variable x que puede tomar valores en el intervalo $[0, V_m]$, donde V_m es la máxima velocidad que puede alcanzar el motor. Se definen entonces tres conjunto difusos lenta, media y rápida en el rango ya definido. Si se toma x como variable lingüística, entonces ella puede tomar lenta, media y rápida como sus valores, de tal manera que se podrá decir que x es lenta, x es media, x es rápida. Lo anterior es presentado en la figura 2.4.

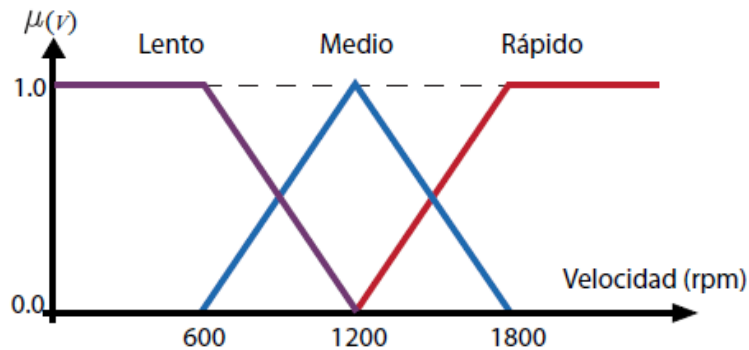


Figura 2.4 Valores que puede tomar la variable *Velocidad* en un motor eléctrico.

Una variable lingüística está caracterizada mediante unos parámetros (X, T, U, M):

X: es el nombre de la variable lingüística.

T: es el conjunto de valores lingüísticos que X puede tomar.

U: es el dominio físico de la variable dentro del cual la variable lingüística X puede tomar cualquier valor cuantitativo.

M: es una regla semántica que relaciona cada valor lingüístico en T con conjunto difuso en U.

De manera que para el ejemplo anterior, se puede definir que X es la velocidad del motor eléctrico, T está conformado por lenta, media y rápida, U es el intervalo $[0, V_m]$ y M es la regla semántica que relaciona lenta, media y rápida con las funciones de pertenencia de la figura 2.4.

2.5. Reglas base

En sistemas difusos el conocimiento humano es representado mediante un conjunto de reglas del tipo IF-THEN. Una regla difusa es una declaración condicional expresada como:

$$\text{IF } \langle \text{antecedente} \rangle \text{ THEN } \langle \text{consecuencia} \rangle$$

Donde la primera proposición es llamada el antecedente o premisa y la segunda es llamada el consecuente o conclusión. La premisa puede contener una combinación de proposiciones mediante conectivos lógicos AND-OR, siendo posible también, que una proposición difusa está basada en la negación como por ejemplo:

$$\text{IF } \langle x \text{ es } A \rangle \text{ AND } \langle y \text{ no es } B \rangle \text{ THEN } \langle z \text{ es } C \rangle$$

Donde A, B y C son valores lingüísticos que pueden tomar las variables x , y y z respectivamente. Existen muchos ejemplos de reglas difusas en la vida diaria tales como:

Si la presión es alta **entonces** el volumen es pequeño.
 Si la carretera es mala **entonces** conducir es peligroso.
 Si la velocidad es alta **entonces** aplique un poco el freno.

A partir de un grupo de estas reglas IF-THEN, está conformada la base de reglas difusas, la cual es el corazón del sistema difuso en el sentido de que todos los componentes son utilizados para implementar estas reglas de una manera eficiente y razonable.

2.6. Operaciones básicas entre conjuntos difusos

2.6.2. Complemento

El complemento del conjunto difuso A es un conjunto \bar{A} cuyo valor de pertenencia es definido como:

$$\mu_{\bar{A}}(X) = 1 - \mu_A(X)$$

Una representación gráfica de esta operación es mostrada en la figura 2.5.

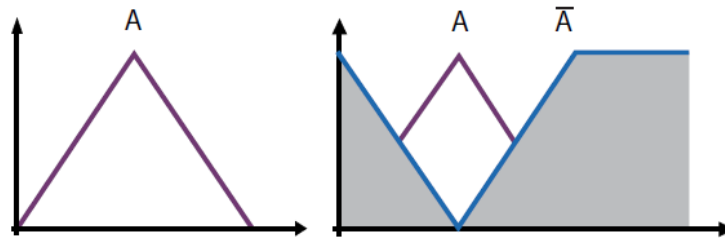


Figura 2.5 complemento de un conjunto difuso

2.6.2. La unión de dos conjuntos difusos.

La unión de dos conjuntos difusos A y B ($A \cup B$) es el conjunto difuso más pequeño que incluye todos los elementos en A o B o ambos A y B. De esto se establece que la unión es un operador lógico OR. La función de membresía de AUB es:

$$\mu_{A \cup B}(X) = \max(\mu_A(X), \mu_B(X)), x \in X$$

En donde $\max(a, b) = a$ si $a \geq b$, y $\max(a, b) = b$ si $a \leq b$. Esto puede ser visto en la figura 2.6.

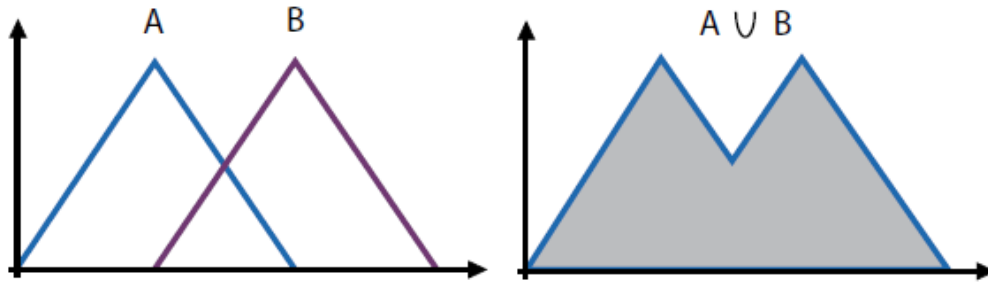


Figura 2.6 Unión entre conjuntos difusos.

2.6.3. La intersección de dos conjuntos difusos

La intersección de A y B ($A \cap B$) es el conjunto difuso más grande dentro de A y B , que indica que la intersección es un operador lógico AND. La función de membresía de ($A \cap B$) es:

$$\mu_{A \cap B}(X) = \min(\mu_A(X), \mu_B(X)), x \in X$$

En donde $\min(a, b) = a$ si $a \geq b$, y $\min(a, b) = b$ si $a \leq b$. Esto puede ser visto en la figura 2.7.

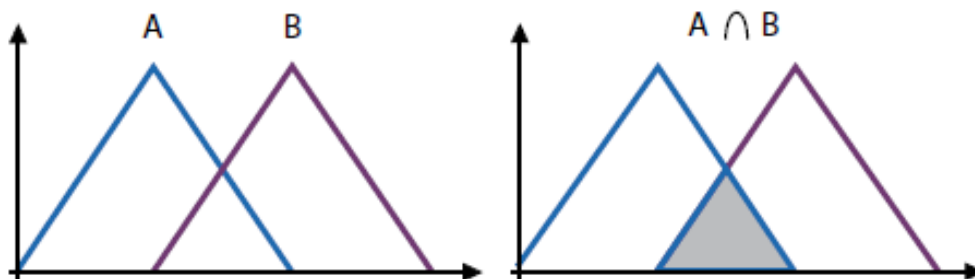


Figura 2.7 Intersección entre conjuntos difusos

La unión y la intersección como operadores máximo y mínimo, son operaciones específicas dentro de dos planteamientos más generales como lo son la norma-T y la norma-S respectivamente. Otros planteamientos tales como los de Yager, Dubois y Prade, entre otros, pueden ser revisados en [1].

2.7. Sistemas de control basados en lógica difusa

La aplicación de la lógica difusa a los sistemas de control, se pueden expresar como un control a través de palabras que interpretan el sentido común en lugar de números o bien sentencias en lugar de ecuaciones, con el fin de incorporar esquemas del razonamiento

humano hacia una estrategia de control automático y aprovechar algunas veces la experiencia del operador del sistema.

El controlador basado en lógica difusa, es utilizado para sistemas de control en lazo cerrado, en los cuales la señal de control es una decisión tomada a partir de la información de que tan diferente es el comportamiento de la planta con respecto al deseado, es decir, a partir de la señal de error. En la figura 2.8 se muestra la estructura de control en lazo cerrado con controlador basado en lógica difusa



Figura 2.8 Sistema de control difuso en lazo cerrado.

La lógica difusa logra interpretar las sentencias del lenguaje común fundamentado sobre un soporte matemático que permite extraer conclusiones cuantitativas a partir de un conjunto de observaciones y reglas cualitativas. Sin embargo, las variables de los procesos no se miden ni se aplican a los actuadores en sentido común sino en números. Con el fin de resolver este inconveniente, se hace necesario utilizar un esquema que convierta las señales numéricas del mundo real en variables difusas como una adaptación previa antes de introducir el estado de una variable al controlador y también uno que convierta las variables difusas en números para que la señal pueda ser aplicada a la planta. La figura 2.9 muestra una estructura típica de un controlador basado en lógica difusa.

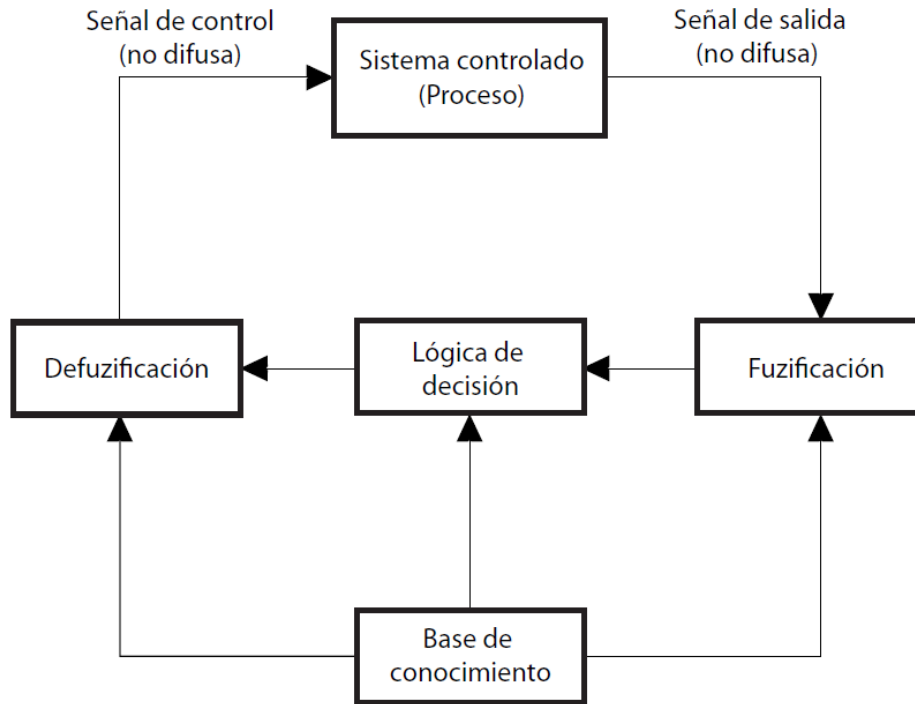


Figura 2.9 Configuración del sistema difuso

Esta estructura tiene cuatro componentes fundamentales para llevar a cabo la estrategia de control y son las siguientes:

2.7.1. Base de conocimiento

Es el corazón del sistema difuso en el sentido de que contiene información necesaria para el funcionamiento de los otros componentes del sistema. Está formada por una base de datos y una base de reglas. La primera recoge la definición de las funciones de pertenencia de las entradas y salidas y toda la información necesaria para caracterizar las reglas de control y manipulación de datos; y la segunda, es un conjunto de reglas que caracteriza y resumen los objetivos de control de un experto por medio de expresiones lingüísticas de control.

La base de datos está relacionada con los soportes sobre los cuales los conjuntos difusos son definidos. Las funciones de membresía son definidas por algunos parámetros que varían según la forma de la función deseada, obteniendo de esta manera funciones como la mostrada en la figura 2.10

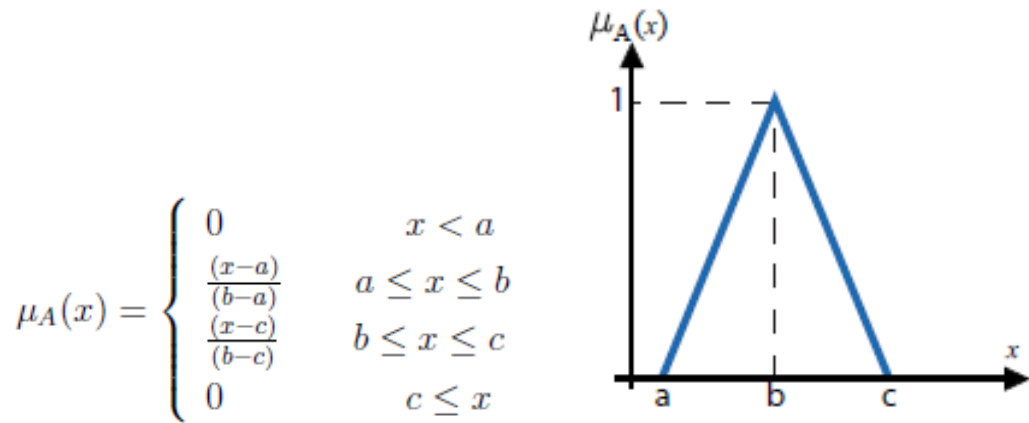


Figura 2.10 Construcción de una función de membresía.

En el momento de definir las funciones de membresía se debe cubrir todo el rango de valores posibles que puede tomar la variable medida o rango de valores que se pueden asignar a la señal que será aplicada al sistema controlado. La selección de la cantidad de conjuntos difusos para representar una variable determina la resolución del control obtenible, pues el mismo espacio se puede definir con distinto número de conjuntos como se muestra en la figura 2.11, haciendo de esta una de las consideraciones para el diseño en las que más cuidado se debe tener, convirtiéndola en un procedimiento heurístico de ensayo y error donde se busca una partición óptima [7].

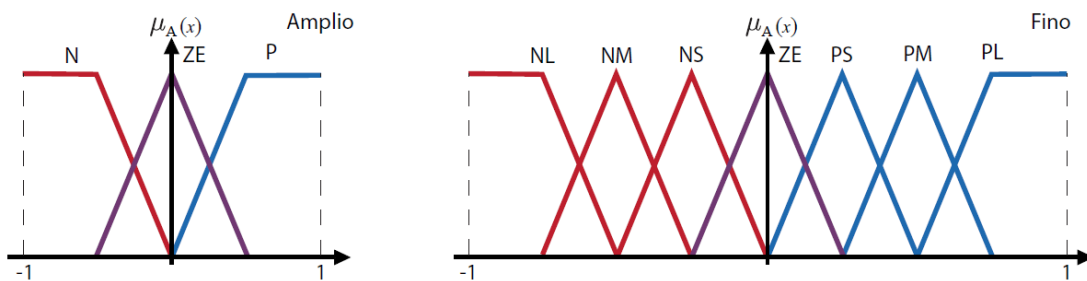


Figura 2.11 Resolución obtenida según el número de funciones de membresía.

La imagen a la izquierda de la figura 2.11 muestra un espacio definido por tres conjuntos difusos:

N: Negativo. ZE: Cero. P: Positivo.

Obteniendo así una resolución amplia, pues cada conjunto difuso está abarcando una gran cantidad de posibles valores que puede asumir la variable en algún momento, mientras que la imagen de la derecha muestra una variable definida por siete conjuntos difusos de la siguiente manera:

NL: Negativo Grande. NM: Negativo Mediano. NS: Negativo Pequeño. ZE: Cero.
 PS: Positivo Pequeño. PM: Positivo Mediano. PL: Positivo Grande.

Debido a que el rango de valores posibles es definido por una mayor cantidad de conjuntos difusos se puede lograr una resolución más fina pero puede traer ciertas dificultades en los procesos de difusión y concreción.

La base de reglas caracteriza y resume los objetivos de control de un experto por medio de un conjunto de reglas lingüísticas basadas en el conocimiento, que suele ser expresado en la forma de reglas IF-THEN, las cuales son fácilmente expresadas mediante declaraciones condicionales como se muestra:

IF T es NL AND P es ZE THEN Y es PL.
IF T es ZE AND P es PM THEN Y es NM.
IF T es PM AND P es NM THEN Y es ZE.

Estas reglas pueden ser presentadas como se mostraron anteriormente o también en una base de datos completos y organizados como se muestra en la figura 2.12, la cual representa una base de reglas para el control de una turbina de vapor que tiene como variables de entrada medidas de temperatura y presión y la salida es la velocidad.

$\begin{matrix} T \\ P \end{matrix}$	NL	NM	NS	ZE	PS	PM	PL
NL	PL	PL	PL	PL	PM	PS	ZE
NM	PL	PL	PM	PM	PS	ZE	NS
NS	PL	PM	PS	PS	NS	NM	NL
ZE	PL	PM	PS	ZE	NS	NM	NL
PS	PL	PM	PS	NS	NS	NM	NL
PM	PM	ZE	NS	NM	NM	NL	NL
PL	ZE	NS	NM	NL	NL	NL	NL

Figura 2.12 Base de reglas difusas.

El cuadro de la figura 2.12 es una tabla de memoria asociativa difusa (MAD) [8], la cual resume todas las reglas que regirán la forma en que el controlador tomará las decisiones acerca de cómo debe ser la señal de control para que el sistema controlado tenga el comportamiento deseado.

2.7.2. Interfaz de difusión (Fuzificación)

Realiza un escalado de los valores de las entradas para adecuarlos a los valores típicos para los que se define el controlador y luego hace la difusión (fuzificación) que convierte los

datos de entrada en valores lingüísticos adecuados para su manipulación. En esta etapa, se le otorga a los datos de entrada un grado de membresía a uno o varios de los conjuntos difusos en los que se divide el rango de valores posibles para dicha entrada, para ello busca la correspondencia entre el estado de las variables y las funciones de membresía definidas para tal propósito.

Para un determinado valor de una variable su grado de pertenencia sería mayor en una expresión que en las demás como se presenta en la figura 2.13, donde se muestra una fuzificación en el instante t_0 de una variable $T(t)$ cuyo universo de discurso está definido por siete funciones triangulares con un traslape del 50% entre funciones.

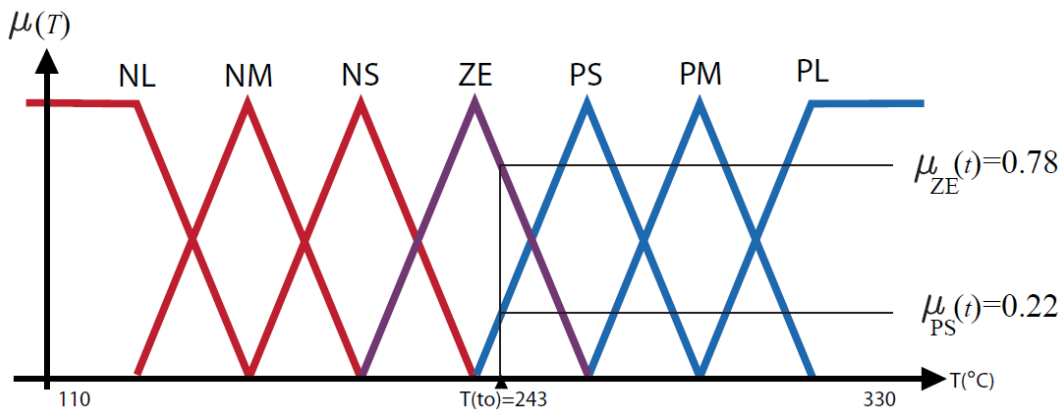


Figura 2.13 Proceso de difusión de una medida de temperatura.

De esta manera se le asigna grados de membresía a los distintos conjuntos difusos que definen el universo de discurso de la señal de entrada que en este caso es una medida de temperatura registrada por un sensor. Así para el caso ilustrado en la figura 2.13, donde la entrada es un registro de temperatura de 243 grados centígrados puede ser expresado de la siguiente manera:

Valores de membresía de la variable temperatura (T):

$$\begin{array}{llll} \mu_{NL}(T_o) = 0,0 & \mu_{NM}(T_o) = 0,0 & \mu_{NS}(T_o) = 0,0 & \mu_{ZE}(T_o) = 0,78 \\ \mu_{PS}(T_o) = 0,22 & \mu_{PM}(T_o) = 0,0 & \mu_{PL}(T_o) = 0,0 & \end{array}$$

Valores de membresía de la variable presión (P):

$$\begin{array}{llll} \mu_{NL}(P_o) = 0,0 & \mu_{NM}(P_o) = 0,0 & \mu_{NS}(P_o) = 0,63 & \mu_{ZE}(P_o) = 0,37 \\ \mu_{PS}(P_o) = 0,22 & \mu_{PM}(P_o) = 0,0 & \mu_{PL}(P_o) = 0,0 & \end{array}$$

A partir de estos valores de membresía asignados a los valores de entrada y haciendo uso de la base de reglas mostradas en la figura 2.12, se puede deducir fácilmente que las reglas necesarias para tomar la decisión en este caso son:

- Regla 1. **IF** T es ZE **AND** P es NS **THEN** V es PS.
 Regla 2. **IF** T es ZE **AND** P es ZE **THEN** V es ZE.
 Regla 3. **IF** T es PS **AND** P es NS **THEN** V es NS.
 Regla 4. **IF** T es PS **AND** P es ZE **THEN** V es NS.

Estas reglas deben ser evaluadas con los valores de membresía haciendo uso del operador AND, a partir del cual se define el valor mínimo entre las dos entradas como valor de membresía al conjunto de la salida como se muestra en la figura 2.14 donde se presenta la evaluación de la Regla 1:

Regla 1. **IF** T es ZE **AND** P es NS **THEN** V es PS.

$$\mu_{PS}(V) = \min\{\mu_{ZE}(T), \mu_{NS}(P)\}$$

$$\mu_{PS}(V) = \min\{0.78, 0.63\} = 0.63$$

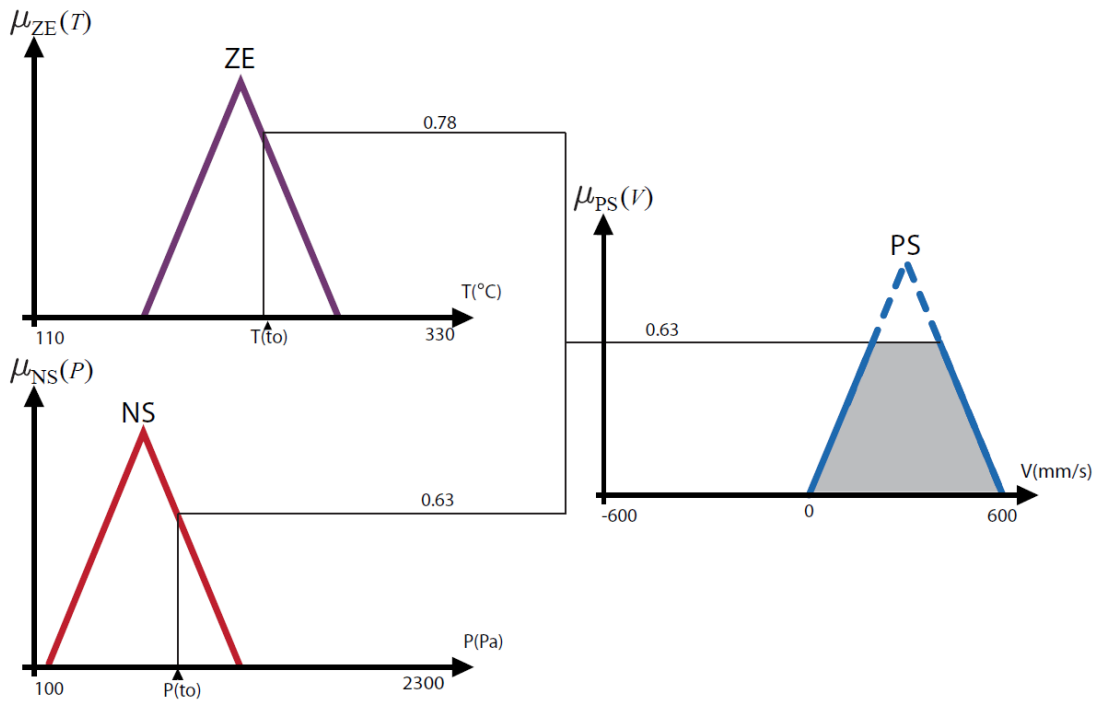


Figura 2.14 Representación gráfica de la evaluación de Regla

En la figura 2.14 se presenta como actúa la lógica de decisión en un sistema con dos entradas (temperatura y presión) en un instante t_0 y asigna el menor grado de membresía de las dos anteriores, como el grado de membresía al conjunto PS de la variable de salida como lo indica la regla. De igual manera se debe realizar la evaluación de cada una de las reglas involucrada en el proceso como se muestra:

$$\text{Regla 2. } \mu_{ZE}(V) = \min\{0.78, 0.37\} = 0.37$$

$$\text{Regla 3. } \mu_{NS}(V) = \min\{0.22, 0.63\} = 0.22$$

$$\text{Regla 2. } \mu_{NS}(V) = \min\{0.22, 0.37\} = 0.22$$

Todo este procedimiento realizado utilizando la lógica de decisión es mostrada en la figura 2.15, en el cual se presenta como a partir de la información de las dos variables de entradas y evaluando cada una de las reglas involucradas en este proceso de decisión se obtiene un área y unos conjuntos asignado para la variable de salida.

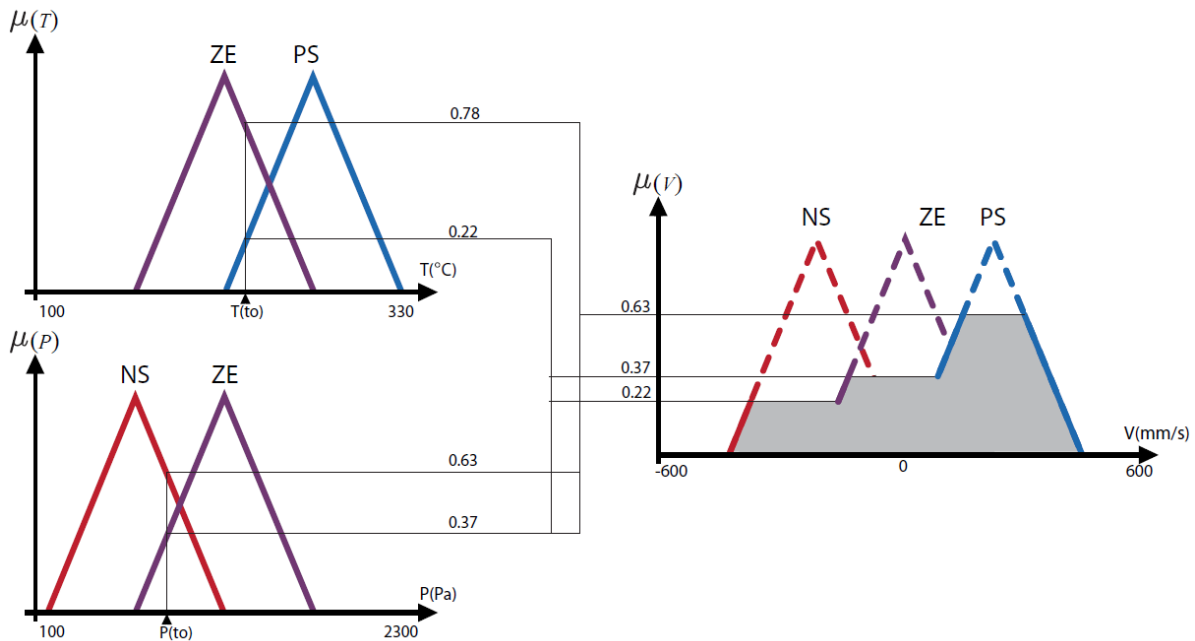


Figura 2.15 Lógica de decisión.

Pueden considerarse como criterios para la selección, factores tales como los motores de inferencia, normas t y normas s , técnicas de inferencia, la apreciación intuitiva, la eficiencia computacional y las propiedades especiales que presentan algunos mecanismos de inferencia, tales como inferencia de producto, inferencia mínimo, inferencia de Zadeh, entre otros que son muy usados en sistemas difusos y de control que son presentados en [6].

2.7.3. Interfaz de concreción (Defuzificación)

Los sistemas que actúan en el mundo real requieren de señales definidas dentro de los números reales, por lo que es necesario convertir las señales entregadas a las salidas de los sistemas difusos a números reales. La interfaz de concreción convierte la acción difusa inferida en el proceso de la lógica de decisión, en una acción concreta susceptible de

aplicación sobre el proceso, y donde se realiza un escalamiento para adecuar los rangos de salida a los que se ha definido el controlador con las entradas del proceso.

Para el ejemplo de la turbina de vapor se podría decir mediante la evaluación de reglas que la velocidad debe ser Positivo Pequeño pues en los resultados de la lógica de decisión tiene la mayor membresía, pero esto no es suficiente, ya que es preciso obtener un valor numérico que indique la velocidad exacta a la que debe girar la turbina.

El método de concreción más utilizado es el Método del centro de gravedad, el cual consiste en crear para la salida del sistema una función de membresía a un nuevo conjunto obtenido como unión de aquellos a los que pertenece parcialmente el valor de salida como se muestra en la figura 2.16.

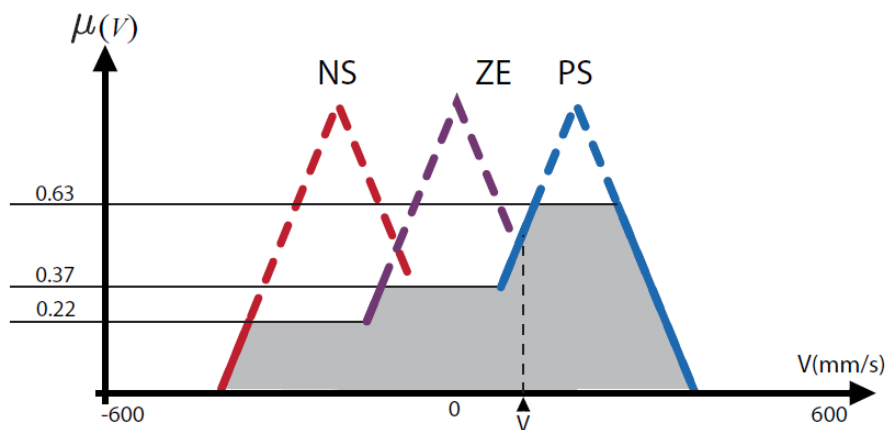


Figura 2.16 Concreción utilizando método del centro de gravedad

Se debe localizar el centro de gravedad de la figura obtenida, el cual definirá cual es el valor de salida del sistema de control que será aplicado al sistema controlado para que este último permanezca en el punto de operación deseado. La ordenada de la ubicación del centro de gravedad para la figura 2.16 se obtiene del siguiente modo.

$$CG = \frac{\sum_j v_j * \mu_V(v_j)}{\sum_j \mu_V(v_j)}$$

El centro de gravedad (CG) está definido como la suma de los productos resultante entre el centro horizontal del conjunto j y el grado de membresía a dicho conjunto, dividida por la suma de los valores de membresía resultantes de la lógica de decisión.

$$CG = \frac{(0.63)(200) + (0.37)(0) + (0.22)(-200) + (0.22)(-200)}{0.63 + 0.37 + 0.22 + 0.22}$$

$$CG = V = \frac{38}{1.44} = 26.39$$

Este resultado obtenido, significa que la acción de control debe ser de 26,39 mm/s para lograr mantener el sistema en la posición deseada.

Se proponen distintos métodos para la concreción tales como el promedio de centros, centro de sumas, promedio de los máximos, centro máximo, primero o último de máximos, centro de área, centro del área más grande, altura y máxima altura, los cuales pueden ser revisados en [6] y [9].

El criterio más difundido para la selección del método de defuzificación es la simplicidad computacional debido a que no hay bases científicas (ninguno de ellos ha sido derivado de un principio tal como la maximización de la información difusa).

SISTEMAS NEURO DIFUSO

3.1. Introducción

El mayor reto de la ciencia moderna es desarrollar trabajos de investigación enfocados en las máquinas para realizar procesos de forma inteligente, y creando algoritmos avanzados de inteligencia artificial, haciendo que la programación de estos sistemas sea más compleja pero buscando más aplicaciones que respondan a las necesidades humanas, pareciéndose más al pensamiento humano. Con el rápido desarrollo de las técnicas de redes neuronales y sistemas de lógica difusa, los sistemas neuro-difusos están atrayendo cada vez más interés ya que son más eficientes y más poderosos que cualquiera de las redes neuronales o sistemas de lógica difusa. Sin embargo, se ha hecho el progreso más importante en los sistemas neuro-difusos en los últimos años. La mayoría de los trabajos de investigación y aplicaciones aparecieron en la década de 1990. Desde el punto de vista teórico, muchos algoritmos de aprendizaje eficaces para sistemas neuro-difusos se desarrollaron y se propusieron muchas estructuras. Por ejemplo, los sistemas de inferencia de Jang adaptativos basados en red-difusos, sistema de control de lógica difusa y la decisión basada en redes neuronales de Lin, varios sistemas de Wang adaptativos difusos, la ARTMAP difusa por Carpenter, las difusas redes de clustering Kohonen por Bezdek, la red neuronal difusa con señales difusas y pesos por Hayashi, la red neuro-difusa con entradas difusas y objetivos difusos por Ishibuchi, la red neuro-difusa aprender las reglas de control difusos y funciones de pertenencia del error por backpropagation por Nauck y Kruse, etc. En el aspecto de aplicaciones, sistemas neurodifusos han sido ampliamente utilizados en sistemas de control, el reconocimiento de patrones, productos de consumo, medicina, sistemas expertos, matemática borrosa, la teoría de juegos, etc. En el aspecto interdisciplinario, cada vez más otras técnicas, tales como los algoritmos genéticos y el caos, se están aplicando para mejorar los sistemas neuro-difusos. [10]

Los algoritmos de inteligencia artificial (IA), se han aplicado a los sistemas de control para que estos respondan más rápido, ya que la inexactitud de los modelos matemáticos de las plantas desmejoran el rendimiento del controlador especialmente para los problemas de control no lineal o complejos, y con las IA se proporciona una memoria a la cual la máquina aprende en cada proceso de la iteración del algoritmo.

El enfoque de los sistemas Neuro-Difusos se presenta como el reconocimiento de patrones, la adaptación que se hace frente a los cambios del entorno, esto como tal es el funcionamiento de las redes neuronales, y en la toma de decisiones, el razonamiento inferencial y el conocimiento humano son los parámetros que se tiene en cuenta en la lógica difusa. La integración de estos dos enfoques complementarios y junto con cierta técnica de

optimización en derivación, tiene la ventaja de que los parámetros del sistema tienen un significado físico, el cual se observa desde el modelo matemático hasta en el momento en el cual se aplica en las diferentes técnicas de control.

Para realizar la estructura de un sistema Neuro-Difuso, se debe tener expresado las reglas lingüísticas y así poder construir principalmente el sistema Difuso y también se dispone de cierta cantidad de datos o si se ha establecido ya un tipo de entrenamiento o aprendizaje; al tener esto en cuenta se puede establecer una estructura de construcción para la RNA y especificar el algoritmo de aprendizaje. [11]

Las taxonomías de los sistemas neuro-difusos se caracterizan por las diferentes combinaciones que pueden ser remarcados hasta el día de hoy como son: los modelos concurrentes de los cuales las redes neuronales y los sistemas difusos trabajan juntos pero sin que alguno determine las características del otro, por ende las redes neuronales asisten a los sistemas difusos o viceversa, esto solo se presenta cuando las variables del controlador no pueden ser medidas directamente, en algunos casos las salidas de los sistemas de inferencia difusa no pueden ser aplicadas directamente al proceso, en este caso las redes neuronales pueden actuar como un post proceso. Otro modelo es el cooperativo del cual consiste en que las redes neuronales se usan para determinar las funciones de membresía o para determinar las reglas difusas a partir de los datos de entrenamiento (en otras palabras las redes neuronales funcionan como un procesador), y para este modelo se describen dos fases, la primera la de entrenamiento de la cual las redes neuronales interactúan con el sistema difuso para determinar los parámetros del mismo; y la segunda, las redes neuronales desaparecen dejando solo al sistema difuso. El último modelo está basado en que tanto las redes neuronales y los sistemas difusos trabajen juntos en una arquitectura homogénea que puede ser interpretada como una red neuronal con parámetros difusos o como un sistema difuso con parámetros o funcionamiento distribuidos, se conoce como modelo híbrido. [12]

3.2. MÉTODO DEL GRADIENTE

Se recordará que el gradiente es un vector en un punto x que proporciona la dirección (local) de máxima variación de la función. El vector gradiente es un vector ortogonal al contorno de la función en el punto. Por lo tanto en la búsqueda de un mínimo la dirección de movimiento será contra gradiente:

$$s^k = -\nabla f(x) \quad (3.1)$$

En el método de máximo descenso la transición de un punto x^k a otro x^{k+1} viene dada por la siguiente expresión:

$$x^{k+1} = x^k + \Delta x^k + \lambda^k s^k = x^k - \lambda^k \nabla f(x^k) \quad (3.2)$$

Dónde: $\Delta x^k =$ Vector desde x^k hasta x^{k+1}

s^k = Dirección de búsqueda de máximo descenso

λ^k = Escalar que determina la longitud de paso en la dirección s^k

El gradiente negativo da la dirección de movimiento, pero no la longitud de dicho movimiento. Por lo tanto existen varios procedimientos posibles dependiendo de la elección de λ^k . Entre los diversos métodos que existen para la selección de la longitud de paso, dos merecen una mención especial. El primero de ellos emplea una búsqueda unidireccional en la dirección del gradiente. El segundo especifica a priori la longitud del paso para cada iteración. Claramente la principal dificultad con la segunda aproximación es que a menudo no se sabe a priori la longitud de paso adecuada.

Si en cada paso se realiza una optimización total en la dirección contraria al gradiente, los pasos sucesivos del método de máximo descenso son ortogonales uno con respecto al anterior. Este resultado, que parece peculiar, ocurre para una determinada $f(x)$ debido a que la derivada de $f(x)$ a lo largo de la línea $s(\lambda)$ viene dado, utilizando la regla de la cadena por:

$$\frac{df}{d\lambda} = \sum_i \frac{dx_i(\lambda)}{d\lambda} \frac{df}{dx_i} = \sum_i s_i \frac{\partial f}{\partial x_i} = \mathbf{s}^T \nabla f \quad (3.3)$$

en el paso final de la búsqueda se quiere que $\frac{df}{d\lambda} = \mathbf{0}$ y por lo tanto $\mathbf{s}^T \nabla f(\mathbf{x}^{k+1})$. En la práctica, por lo tanto no es deseable llevar a cabo suboptimizaciones con demasiada precisión. Mientras que el método del gradiente puede producir progresos muy satisfactorios en la reducción de $f(x)$ en la primeras iteraciones, tiende a hacerse muy lento en las últimas. Alargando excesivamente los cálculos. El algoritmo práctico se puede resumir en los siguientes pasos:

- 1.- Elegir un valor inicial \mathbf{x}^0 . En pasos sucesivos será \mathbf{x}^k .
- 2.- Calcular, analítica o numéricamente las derivadas parciales $\frac{\partial f}{\partial x_j}$ $j = 1, 2, 3, \dots, n$.
- 3.- Calcular el vector de búsqueda $\mathbf{s} = -\nabla f(\mathbf{x}^k)$
- 4.- Usar la relación $\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda^k \mathbf{s}^k$ para obtener el siguiente punto. El valor de λ^k puede ser de valor fijo o calculado en cada paso mediante una búsqueda unidireccional.
- 5.- Comparar $f(\mathbf{x}^{k+1})$ con $f(\mathbf{x}^k)$. Si el cambio es menor que una tolerancia preespecificada terminar, en caso contrario volver al paso dos y continuar con las iteraciones.

Un método estricto de descenso máximo puede terminar en cualquier punto estacionario, es decir, puede llegar a un mínimo local o a un punto de silla. Para asegurar que tipo de resultado se ha obtenido se debe asegur que la matriz Hessiana es definida positiva. Por otra parte la dificultad básica del método de gradiente es que es muy sensible al escalado de $f(x)$ por lo que la convergencia puede ser muy lenta y producirse un número enorme de oscilaciones. Desde este punto de vista el método del gradiente no es muy efectivo. [13]

3.2.1. MÉTODO DEL GRADIENTE CONJUGADO

El método del gradiente conjugado debido a Fletcher y Reeves (1964) [4] combina las características de la convergencia cuadrática del método de las direcciones conjugadas con las del método del gradiente. El método supone una importante mejora del método del gradiente con sólo un pequeño incremento en el esfuerzo de cálculo. El método del gradiente conjugado, esencialmente, combina la información obtenida del vector gradiente con la información acerca del vector gradiente de iteraciones previas. Lo que hace el método es calcular la nueva dirección de búsqueda utilizando una combinación lineal del gradiente en la etapa considerada y el de la etapa anterior. La principal ventaja del método es que necesita almacenar muy poca cantidad de información con lo que puede ser programado fácilmente incluso en calculadoras.

Los pasos de cálculo se comentan a continuación:

- 1.- En \mathbf{x}^0 (punto inicial) calcular $f(\mathbf{x}^0)$ y calcular $\mathbf{s}^0 = -\nabla f(\mathbf{x}^0)$.
- 2.- Almacenar $\nabla f(\mathbf{x}^0)$ y calcular $\mathbf{x}^1 = \mathbf{x}^0 + \lambda^0 \mathbf{s}^0$ minimizando λ mediante una búsqueda unidireccional en la dirección \mathbf{s}^0 .
- 3.- Calcular $f(\mathbf{x}^1)$, $\nabla f(\mathbf{x}^1)$ la nueva dirección de búsqueda es una combinación lineal de \mathbf{s}^0 y $\nabla f(\mathbf{x}^1)$:

$$\mathbf{s}^1 = -\nabla f(\mathbf{x}^1) + \mathbf{s}^0 \frac{\nabla^T f(\mathbf{x}^1) \nabla f(\mathbf{x}^1)}{\nabla^T f(\mathbf{x}^0) \nabla f(\mathbf{x}^0)} \quad (3.4)$$

para la etapa k-ésima la relación es:

$$\mathbf{s}^{k+1} = -\nabla f(\mathbf{x}^{k+1}) + \mathbf{s}^k \frac{\nabla^T f(\mathbf{x}^{k+1}) \nabla f(\mathbf{x}^{k+1})}{\nabla^T f(\mathbf{x}^k) \nabla f(\mathbf{x}^k)} \quad (3.5)$$

Para una función cuadrática se puede demostrar que dos direcciones de búsqueda son conjugadas. Después de n iteraciones conviene comenzar otra vez desde el principio tomando el último punto $k=n$ como nuevo punto de partida.

- 4.- Realizar el test de convergencia, (la función objetivo ha disminuido), y terminar el algoritmo cuando $\|\mathbf{s}^k\|$ sea menor que alguna tolerancia preestablecida. Algunas de las dificultades que aparecen en el método de Powell también aparecen en el método del gradiente conjugado. Se puede producir una dependencia lineal de las direcciones de búsqueda. Esto se puede eliminar reiniciando el proceso cada cuatro o cinco etapas completas. [13]

3.3. ESTRUCTURA DEL SISTEMA NEURO-DIFUSO

Con la forma de iniciar la estructura base del sistema, se establece por medio del conjunto de reglas **SI-ENTONCES**, se indican tanto las variables a estudiar como los parámetros a definir.

$$\text{SI } x_1 \text{ es } A_{i1} \text{ Y ... Y } x_n \text{ es } A_{in}, \text{ ENTONCES } y_i \text{ es } m_i \quad (3.6)$$

Sea $\mu^{(k)}$ y $a^{(k)}$ como denotaciones de las entradas y salidas de un nodo en la capa k, respectivamente. Las funciones de los nodos en cada una de las capas del sistema neuro-difuzo se describen desde la ecuaciones (3.7) a la (3.15)

En la Capa 1 no se realiza ningún cálculo, ya que cada nodo en esta capa que corresponde a una sola variable de entrada, transmite los valores de entrada a la siguiente capa directamente. Cada nodo tiene una función de membresía, que

$$\begin{aligned} a_i^1 &= \mu_{A_i}(x) \\ a_j^1 &= \mu_{B_j}(x) \end{aligned} \quad (3.7)$$

especifica el grado del cual x satisface A_i , y se puede establecer con cualquier función de pertenencia con parámetros apropiados, tales como la función de una campana Gaussiana, ver ecuación (3.8)

$$\mu_{A_i}(x) = \mu_{B_i}(x) = \exp \left[- \left(\frac{x-m_i}{\sigma_i} \right)^2 \right] \quad (3.8)$$

Donde σ_i , m_i son, respectivamente, el centro (o media) y la anchura (o la varianza) de la función de pertenencia gaussiana. El centro y la anchura se determinan de la de la ecuación (3.9) y (3.10) respectivamente.

$$m_i = \frac{(i-4)}{3} NRME \quad (3.9)$$

$$\sigma_i = \frac{NRME}{9\sqrt{\ln(2)}} \quad (3.10)$$

Donde NRME es el rango del error.

En la Capa 2 se está tomando como entrada los valores de salida que se presenta en la Capa 1. Se resalta que cada nodo está representado como una regla de lógica difusa, de acuerdo a cuantas funciones de membresías se establecieron previamente y así se estipula los números de neuronas que se implementa en esta capa. Aquí se utilizan diferentes operaciones que

establece la lógica difusa como AND (min: intersección) o OR (max: unión); como se describe en la sección 2.6. y representa en la ecuación (3.11)

$$a^{(2)} = \prod \mu_{A_i}^{(2)} * \mu_{A_j}^{(2)} \quad (3.11)$$

$$\mu_{B_{ij}} = a^{(2)} \quad (3.12)$$

Donde $\mu_{B_{ij}}$ es la función de membresía de salida en cual está asociada con las reglas de SI y ENTONCES

En la capa 3 se realiza la operación de defuzificación usando el método del centro del área (sección 2.7.3). Los enlaces de los pesos w_{ij} entre el i-esimo nodo de las reglas y el j-esimo nodo de la salida, representan los centros de gravedad de la memoria asociativa Neuro difusa (NFAM) como lo describe la ecuación (3.14).

$$a^3 = \frac{\sum_i \sum_j w_{ij} a^{(2)}}{\sum_i \sum_j a^{(2)}} \quad (3.13)$$

$$a^3 = \frac{\sum_i \sum_j w_{ij} \mu_{B_{ij}}}{\sum_i \sum_j \mu_{B_{ij}}} \quad (3.14)$$

$$a^3 = y \quad (3.15)$$

Donde y es la salida actual de la red neuro-difusa, en la cual, esta salida es una de las variables que se desea implementar en el controlador, siendo esta la señal de control actual.

3.4. ALGORITMO DE APRENDIZAJE USANDO GRADIENTE DESCENDIENTE (Backpropagation)

EL ajuste de parámetros para la red propuesta puede dividirse en dos categorías basados en la parte de IF (antecedentes) y la parte THEN (consecuentes) de las reglas difusas. En la parte de la premisa se realiza el ajuste fino del centro y ancho de la función campana de Gauss, y en la parte de los consecuentes se ajustan los pesos que conectan a los nodos de salida.

Una vez la red ha sido inicializada, se aplica al algoritmo de propagación hacia atrás para ajustar los parámetros haciendo uso de los patrones de entrenamiento. El objetivo principal es minimizar la función error dada por la ecuación (3.16)

$$E = \frac{1}{2} (\dot{u}_T - \dot{u})^2 \quad (3.16)$$

Donde \dot{u}_T es la señal de control derivada y la cual es generada por el entrenador PI y \dot{u} es la señal de control de la planta que la produce el controlador de la red neuro-difusa. La minimización de la función error corresponde a llevar la red hacia un punto donde se obtiene un mínimo global.

Para poder demostrar la forma de propagación el error hacia atrás y encontrar las variaciones correspondientes de los parámetros de interés, es importante primero definir de forma general el contenido de las neuronas que conforman la capa del controlador.

Para cada conjunto de datos de entrenamientos, se realiza un recorrido directo partiendo desde los nodos de entrada, hasta calcular los niveles alcanzados en todos los nodos de la red. Luego, y comenzando por los nodos de salida, se realiza un recorrido hacia atrás para calcular $\partial E / \partial y$. En todos los nodos de las capas ocultas. Suponiendo que x es el parámetro ajustable en el nodo, la regla general de aprendizaje puede expresarse como se muestra en la ecuación (3.17)

$$x_i(t+1) = x_i(t) + \eta \left(-\frac{\partial E}{\partial x_i} \right) \quad (3.17)$$

Donde x_i es el parámetro a ser actualizado y η es el factor de aprendizaje. Mediante la aplicación de la regla de la cadena, se calcula el término de error para cada capa. Luego se realiza la adaptación de los pesos de las capas correspondientes. Después se deriva la ley de aprendizaje para cada capa en la dirección hacia atrás.

$$E = 1/2(\dot{u}_{Tj}^{(3)} - \dot{u}_j^{(3)})^2 \quad (3.18)$$

Para la capa 3 se determinan los valores que se muestran en las ecuaciones (3.19) y (3.20)

$$-\frac{\partial E}{\partial(\text{ent}_j^{(3)})} = -\frac{\partial E}{\partial y_j^{(3)}} \frac{\partial y_j^{(3)}}{\partial(\text{ent}_j^{(3)})} = \frac{(\dot{u}_{Tj}^{(3)} - \dot{u}_j^{(3)})}{\sum_{i=1}^m y_i^{(3)}} = O_i^{(3)} \quad (3.19)$$

$$-\frac{\partial E}{\partial(w_{ij}^{(3)})} = -\frac{\partial E}{\partial y_j^{(3)}} \frac{\partial y_j^{(3)}}{\partial(\text{ent}_j^{(3)})} \frac{\partial(\text{ent}_j^{(3)})}{\partial(w_{ij}^{(3)})} = O_i^{(3)} y^{(2)} \quad (3.20)$$

Los pesos (NFAM) de los consecuentes son actualizados mediante la ecuación (3.21)

$$NFAM_{ij}^{(3)}(t+1) = NFAM_{ij}^{(3)}(t) + \eta \frac{(\dot{u}_{Tj}^{(3)} - \dot{u}_j^{(3)})}{\sum_{i=1}^m y_i^{(3)}} y^{(2)} \quad (3.21)$$

Los rangos del error (NRME) y de la derivada del error (NRMDE) son actualizados mediante las ecuaciones (3.31) y (3.32)

$$-\frac{\partial E}{\partial NRME} = -\frac{\partial E}{\partial y^{(3)}} \frac{\partial y^{(3)}}{\partial NRME} \quad (3.22)$$

$$\frac{\partial E}{\partial y^{(3)}} = (\dot{u}_{Tj}^{(3)} - \dot{u}_j^{(3)}) \quad (3.23)$$

$$\frac{\partial y^{(3)}}{\partial NRME} = \frac{\frac{\partial nduc}{\partial NRME} \frac{nduc}{dduc} \frac{\partial dduc}{\partial NRME}}{dduc} \quad (3.24)$$

$$nduc = \sum_i \sum_j NFAM_{ij}^{(3)} (\prod_{IJ} e^{-x_{ij}}) \quad (3.25)$$

$$dduc = \sum_i \sum_j (\prod_{IJ} e^{-x_{ij}}) = \sum_i \sum_j \left(\prod_{IJ} e^{-\left(\frac{u_j - m_{ij}}{\sigma_{ij}}\right)^2} \right) \quad (3.26)$$

La media y la varianza de la función gaussiana que están en función del rango error y del rango de la derivada del error son deducidas de las ecuaciones (3.9) y (3.10) y se establecen las ecuaciones (3.27) y (2.28)

$$m_e = \frac{(i-4)}{3} NRME, \quad \sigma_e = \frac{NRME}{9\sqrt{\ln(2)}} \quad (3.27)$$

$$m_{\dot{e}} = \frac{(i-4)}{3} NRMDE, \quad \sigma_{\dot{e}} = \frac{NRMDE}{9\sqrt{\ln(2)}} \quad (3.28)$$

$$\frac{\partial nduc}{\partial NRME} = \sum_i \sum_j \frac{2x_j}{\sigma} \left[\frac{j-4}{3} + \frac{x_j}{9\sqrt{\ln(2)}} \right] NFAM_{ij}^{(3)} \prod_{IJ} e^{-x_{ij}} \quad (3.29)$$

$$\frac{\partial dduc}{\partial NRME} = \sum_i \sum_j \frac{2x_j}{\sigma} \left[\frac{j-4}{3} + \frac{x_j}{9\sqrt{\ln(2)}} \right] \prod_{IJ} e^{-x_{ij}} \quad (3.30)$$

$$NRME(t+1) = NRME(t) + \eta (\dot{u}_{T_j}^{(3)} - \dot{u}_j^{(3)}) \frac{\frac{\partial nduc}{\partial NRME} \frac{nduc}{dduc} \frac{\partial dduc}{\partial NRME}}{dduc} \quad (3.31)$$

Se realiza el mismo procedimiento para NRMDE y se obtiene de la ecuación (3.32)

$$NRMDE(t+1) = NRMDE(t) + \eta (\dot{u}_{T_j}^{(3)} - \dot{u}_j^{(3)}) \frac{\frac{\partial nduc}{\partial NRME} \frac{nduc}{dduc} \frac{\partial dduc}{\partial NRME}}{dduc} \quad (3.32)$$

3.5. DISEÑO DEL CONTROLADOR NEURO-DIFUSO

3.5.1. CONDICIONES INICIALES

Los valores iniciales de la red definida son bastante importantes porque puede hacer que con base en una escogencia adecuada el sistema converja rápidamente, o no hacerlo si aquellos están bastante alejados de los parámetros óptimos. Si se requiere una convergencia del sistema debe hacerse una selección inicial de parámetros tal que lo garantice

- **NRME:** Error medio cuadrático sobre todos los patrones de la señal de error, se debe establecer un valor mayor a la señal de error, para que el conjunto difuso reconozca dicho dato.

- **NRMDE:** Error medio cuadrático sobre todos los patrones de la señal derivada del error, al igual que el parámetro anterior, se debe establecer un valor mayor a la señal de la derivada del error, para que el conjunto difuso reconozca dicho dato.

3.5.2. Base de datos

Para el control del sistema fueron seleccionadas como entradas al controlador dos variables que se consideran de gran importancia en el comportamiento del mismo: el error de la velocidad del motor y la derivada del mismo. Se define como salida, la señal de control del entrenador PI y la señal de control derivada del sistema neuro difuso

- **Error:** Es la diferencia entre la señal de entrada (un escalón) y la velocidad. A partir de esta información, el controlador puede saber que tan cerca o alejado se encuentra del punto de operación.
- **Derivada del error:** Representa la tasa de cambio del error y a partir de la cual el controlador conocerá si el error está aumentando o disminuyendo, es decir, si la velocidad se está acercando o alejando del punto de operación.
- **Derivada de la señal de control del entrenador PI (\dot{u}_T):** se establece como la señal de entrenamiento o la señal de salida actual. A partir de las dos entradas se establece la señal de control del entrenador PI, esta se deriva y se establece la señal de error del sistema neuro difuso, como la diferencia de la señal de control del entrenador PI con la salida del sistema neuro difuso.
- **Derivada de la señal de control del sistema neuro difuso (\dot{u}):** Es la salida del sistema neuro difuso y se establece como la señal deseada del sistema, siendo esta la señal que genera un voltaje de entrada a la planta (motor DC). Esta señal está limitada en un rango de [0,5] voltios por protecciones físicas del sistema.

El universo de discurso de cada una de las variables difusas es representado por siete conjuntos difusos con el fin de obtener una buena resolución que logre una fina representación de dichas variables. Estos conjuntos difusos son representados con funciones de membresía gaussianas con un 50% de traslape entre ellas con el fin de lograr simplicidad en el diseño del controlador y utilizar la menor carga computacional posible.

Los conjuntos difusos son divididos en tres grupos Positivo, Negativo y Cero, siendo estos dos primeros subdivididos en Grande, Mediano y Pequeño como se muestra:

- **NB:** Negativo Grande.
- **NM:** Negativo Mediano.
- **NS:** Negativo Pequeño.
- **ZE:** Cero.
- **PS:** Positivo Pequeño.

- **PM:** Positivo Mediano.
- **PB:** Positivo Grande

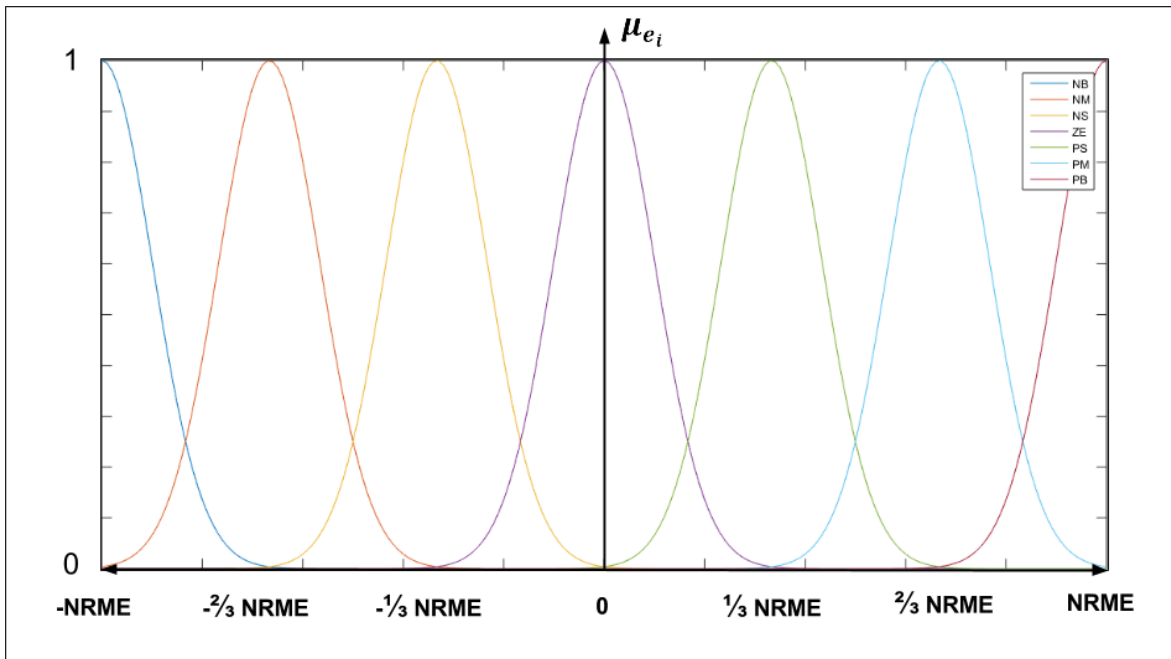


Figura 3.1.a Representación de la variable del error con los conjuntos difusos μ_{e_i}

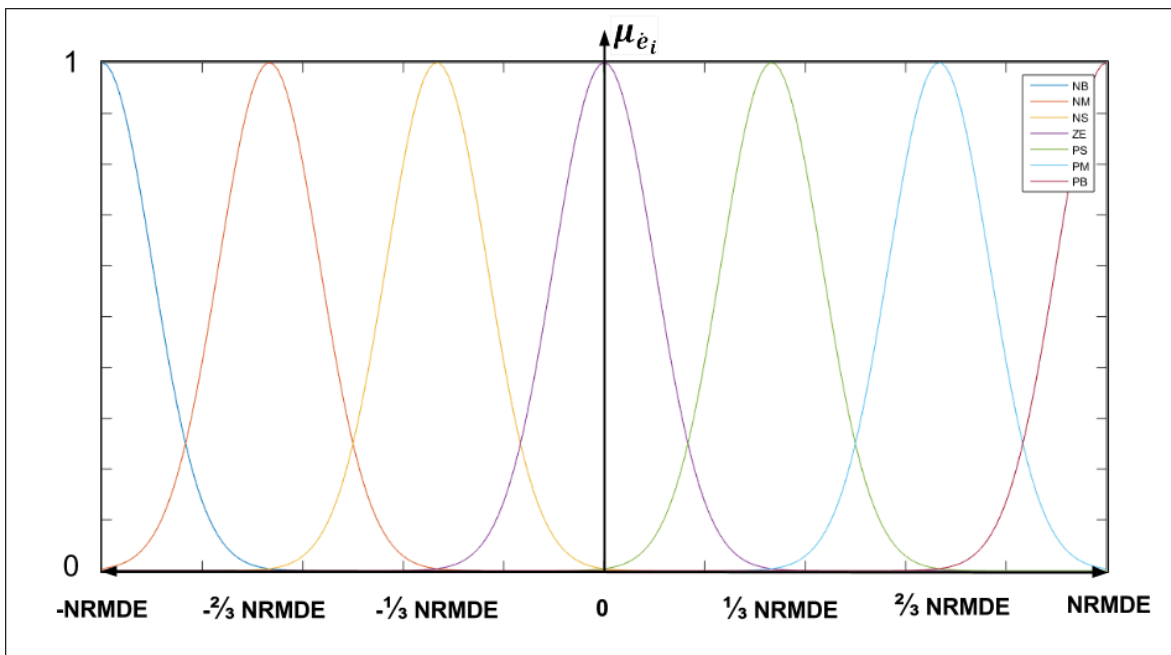


Figura 3.1.b. Representación de la variable derivada del error con los conjuntos difusos $\mu_{\dot{e}_i}$

3.6. Implementación del sistema de control Neuro Difuso

3.6.1. Comportamiento del sistema de control

Durante esta sección se presentan pruebas realizadas al sistema de control Neuro Difuso el cual se entrena en base ne un controlador PI, con la finalidad de observar el comportamiento de la velocidad del motor ante distintos tipos de variaciones de las constantes de proporcionalidad e integral ajustándolas para obtener señales como sobreamortiguado, amortiguado, subamortiguado y oscilatorio; en cada caso también se realiza una perturbación a la planta en cual consiste en establecer un freno al modor DC; debido a que este trabajo pretende mostrar el comportamiento de la planta cuando se encuentra bajo la supervisión del controlador neuro-difuso.

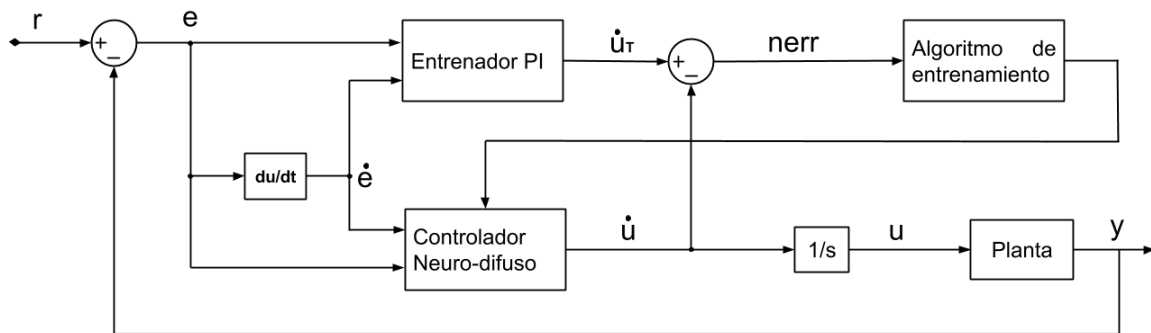


Figura 3.2 Diagrama de bloques del sistema Neuro difuso. Siendo nerr la diferencia entre la señal de control deseada (\dot{u}_r) y la señal de control actual (\dot{u})

3.6.2. Caso sobreamortiguado

En este caso se establece las constantes K_p y K_i en 0.4 y 0.9 respectivamente y se establece un valor de 0.3 para la constante de aprendizaje (η); al inicio de la simulación como se observa en la figura 3.3, muestra el comportamiento de la señal de control de forma sobreamortiguada. Después de un tiempo de entrenamiento se observa en la figura 3.4 en el cambio de la señal de control la variación de los pesos y el ajuste de los rangos de error y la derivada del error; así mismo el funcionamiento de la planta bajo una perturbación como se observa en la figura 3.5.

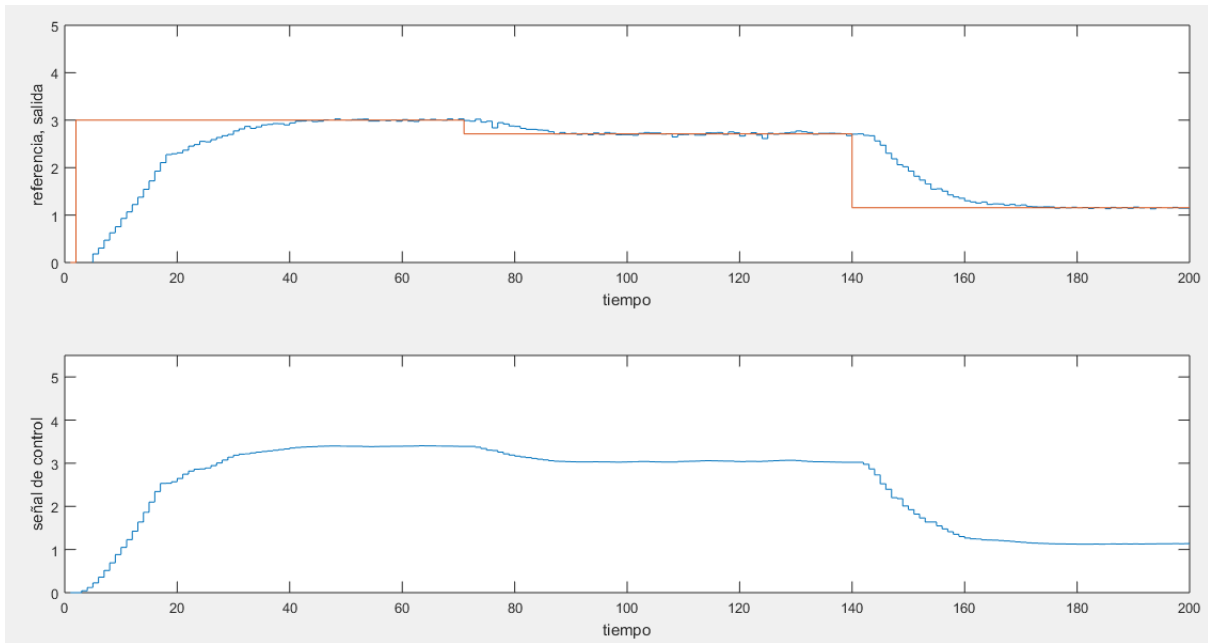


Figura 3.3 Inicio de la prueba para el caso sobreamortiguado. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa

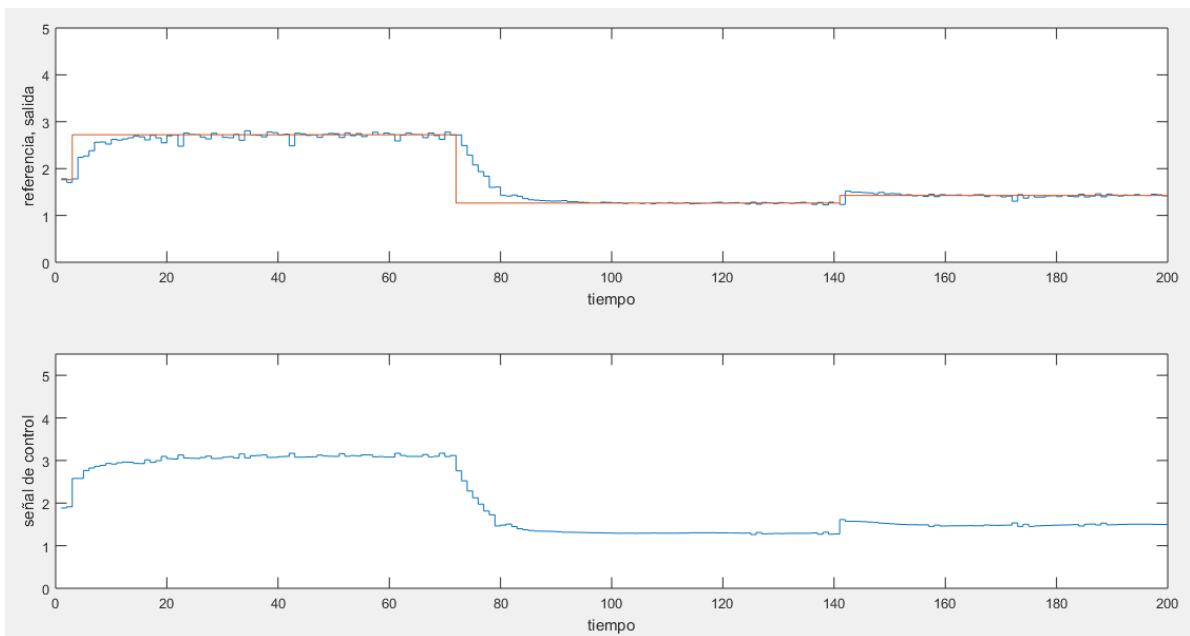


Figura 3.4 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo del cual la señal de salida alcanza la señal de referencia

e/de	NB	NM	NS	ZE	PS	PM	PB
NB	-2.193826777	-0.80472161	-0.536907432	-0.713466716	-0.370185009	-0.254000068	-0.252297895
NM	-1.927927792	-0.28561569	-0.004768113	-0.26023944	-0.031638738	-0.039296142	0.31679924
NS	-0.7159966	-0.41767633	-0.357951628	-0.154674431	-0.070571145	0.0210197	0.243222187
ZE	-0.661459954	-0.26681038	-0.183410582	0.006098609	0.162166131	0.009231253	0.526709594
PS	-0.340296364	-0.03954867	0.046708284	0.101168972	0.304982293	0.343101963	1.399124405
PM	-0.132115128	0.043298246	0.068122714	0.206398415	0.011349781	0.049765127	1.673733069
PB	0.181712288	0.418885058	0.241043686	0.603808959	0.67077523	0.634091158	2.741441844

Tabla 1. Ajuste de los pesos (NFAM) después de un tiempo de iteración. Con variación de los rangos en 1.0078 y 1.1932 para NRME y NRMDE respectivamente y con el error (E) de la diferencia de la señal de control deseada y la señal actual es de 0.0411

Después de haberse generado un entrenamiento adecuado al sistema neuro difuso se inicia los parámetros encontrados de los pesos, los rangos del error y de la derivada del error, y se aplica una perturbación al motor

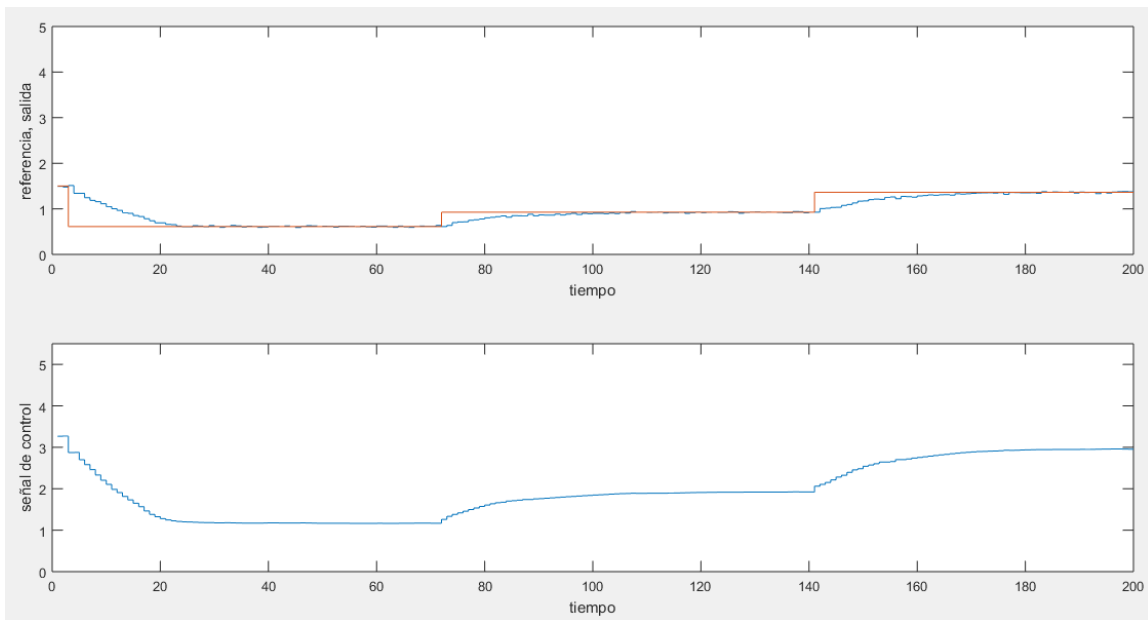


Figura 3.5 Comportamiento de la velocidad del motor bajo una perturbación

e/de	NB	NM	NS	ZE	PS	PM	PB
NB	-2.479492368	-1.00161062	-0.778870169	-0.518518762	-0.290825334	-0.335924516	-0.306051423
NM	-2.181792368	-0.82459422	-0.405020253	-0.207777695	-0.101800733	-0.043085758	0.216
NS	-1.009592368	-0.66739293	-0.505482352	-0.16549828	-0.081427039	0.020991304	0.491199989
ZE	-0.841392368	-0.26069281	-0.200128804	-0.008258142	0.277927536	0.002291881	0.814299989
PS	-0.354692361	-0.03299752	0.003061554	0.251211289	0.475483445	0.449460085	2.102451316
PM	-0.002287798	0.05347656	0.14800599	0.250491019	0.58099986	0.110545564	2.284751328
PB	0.33043084	0.473334446	0.286105111	0.431744339	0.948820943	1.102057741	3.329781568

Tabla 2. Ajuste de los pesos (NFAM) cuando la velocidad está bajo una perturbación. Con variación de los rangos en 0.9910 y 1.3315 para NRME y NRMDE respectivamente y con el error (E) de la diferencia de la señal de control deseada y la señal actual es de 0.0061

3.6.3. Caso amortiguado

En este caso se establece las constantes K_p y K_i en 1 y 2.5 respectivamente, al inicio de la simulación como se observa en la figura 3.6 el comportamiento de la señal de control, de la forma amortiguada. Después de un tiempo de entrenamiento se observa en la figura 3.7 el cambio de la señal de control; así mismo el funcionamiento de la planta bajo una perturbación como se observa en la figura 3.8.

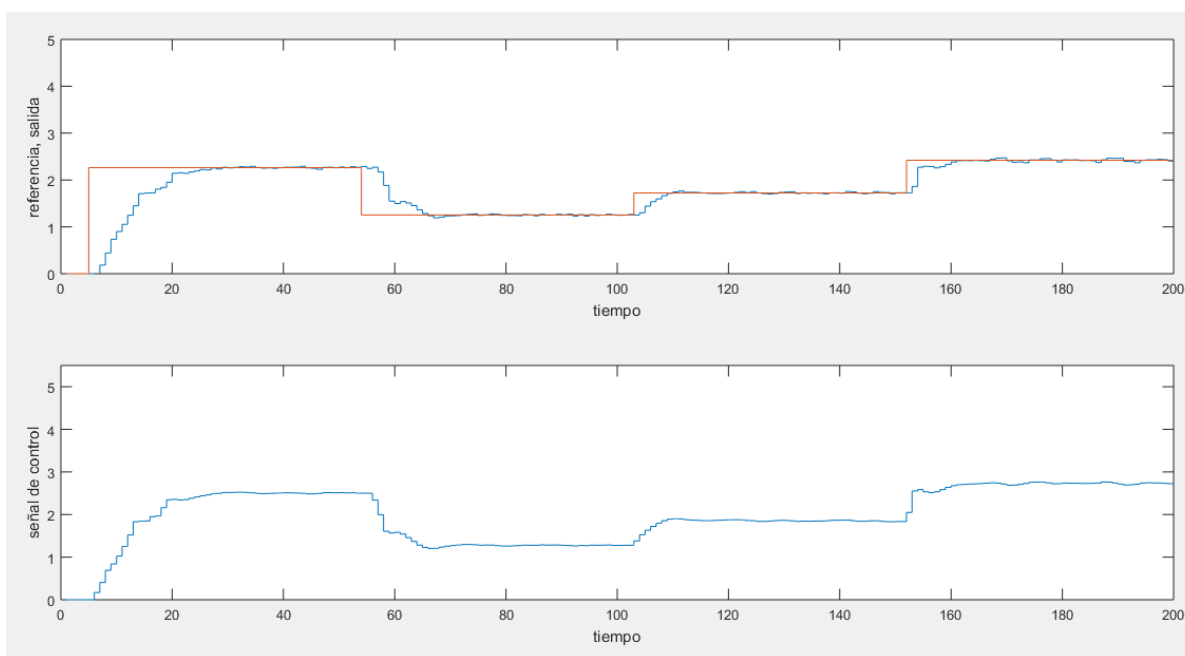


Figura 3.6 Inicio de la prueba para el caso amortiguado. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa.

e/de	NB	NM	NS	ZE	PS	PM	PB
NB	-1.924421071	-0.01307237	-0.014564777	-1.641850697	-0.017561525	0.004264878	-0.039263156
NM	-1.280216592	-1.800142	-2.696901888	-2.155404523	-0.352994873	9.9029E-05	-0.039265945
NS	-1.276516545	-2.26031919	-10.22891403	-3.40846314	0.149790232	-0.000352776	0.499129017
ZE	-0.815108768	-1.05574313	-6.132548867	-0.008916267	5.270713335	0.186006003	1.031543359
PS	-0.77649366	0.049353183	0.591355945	3.441817699	10.42831761	5.26992546	1.032363054
PM	-0.253406052	-0.00197236	0.091284115	1.766985506	0.558314592	1.558808336	1.077780773
PB	0.22912042	0.644002305	1.11989937	1.526957044	0.745057788	0.372290765	9.346371743

Tabla 3. Ajuste de los pesos (NFAM) después de un tiempo de iteración. Con variación de los rangos en 7.7900 y 20.7272 para NRME y NRMDE respectivamente y con el error de la diferencia de la señal de control deseada y la señal actual es de 0.0313

Después de haberse generado un entrenamiento adecuado al sistema neuro difuso se inicia los parámetros encontrados de los pesos, los rangos del error y de la derivada del error, y se aplica una perturbación al motor

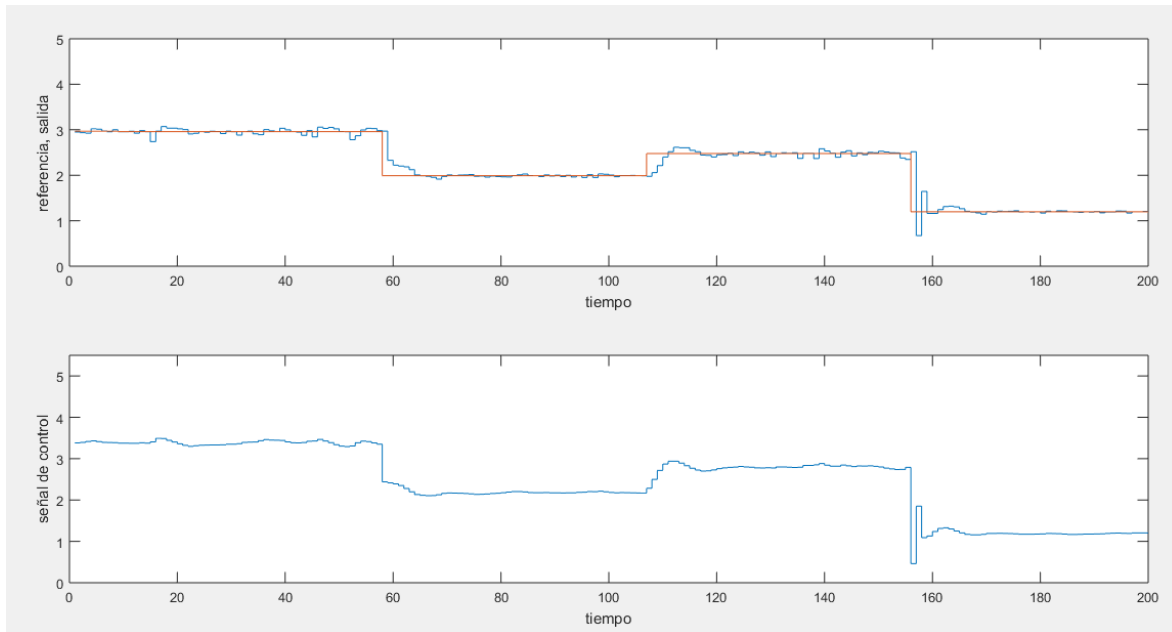


Figura 3.7 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo en el cual la señal de salida alcanza la señal de referencia

Después de haberse generado un entrenamiento adecuado al sistema neuro difuso se inicia los parámetros encontrados de los pesos, los rangos del error y de la derivada del error, y se aplica una perturbación al motor

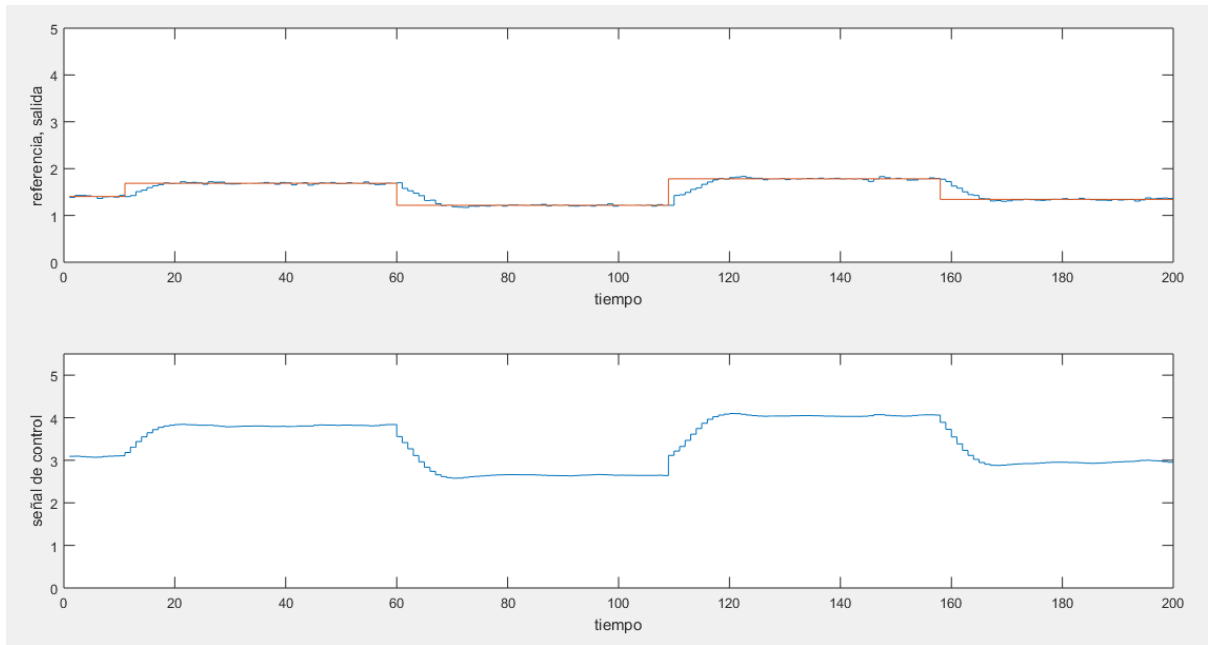


Figura 3.8 Comportamiento de la velocidad del motor bajo una perturbación

e/de	NB	NM	NS	ZE	PS	PM	PB
NB	-1.924421071	-0.01307237	-0.014564777	-1.641850697	-0.017561525	0.004264878	-0.039263156
NM	-1.280216592	-1.800142	-2.696901895	-2.155404529	-0.352994879	9.90291E-05	-0.039265945
NS	-1.276516545	-2.26031077	-10.28297605	-3.461210338	0.152703688	-0.000352558	0.499129017
ZE	-0.815108768	-1.05512331	-5.279914114	-0.048175184	5.409054631	0.186006222	1.031543359
PS	-0.77649366	0.049972998	1.107128205	3.711665643	10.46382779	5.270506618	1.032943993
PM	-0.253406052	-0.00197139	0.091285918	1.766987308	0.558315421	1.57931564	3.166609877
PB	0.22912042	0.644002305	1.11989937	1.526957044	0.745057789	0.392798064	9.375022442

Tabla 4. Ajuste de los pesos (NFAM) cuando la velocidad está bajo una perturbación. Con variación de los rangos en 5.436 y 15.1255 para NRME y NRMDE respectivamente y con el error (E) de la diferencia de la señal de control deseada y la señal actual es de 0.0023

3.6.4. Caso subamortiguado

En este caso se establece las constantes K_p y K_i en 1 y 3.3 respectivamente, al inicio de la simulación como se observa en la figura 3.9 el comportamiento de la señal de control, de la forma subamortiguada. Después de un tiempo de entrenamiento se observa en la figura 3.10 el cambio de la señal de control; así mismo el funcionamiento de la planta bajo una perturbación como se observa en la figura 3.11.

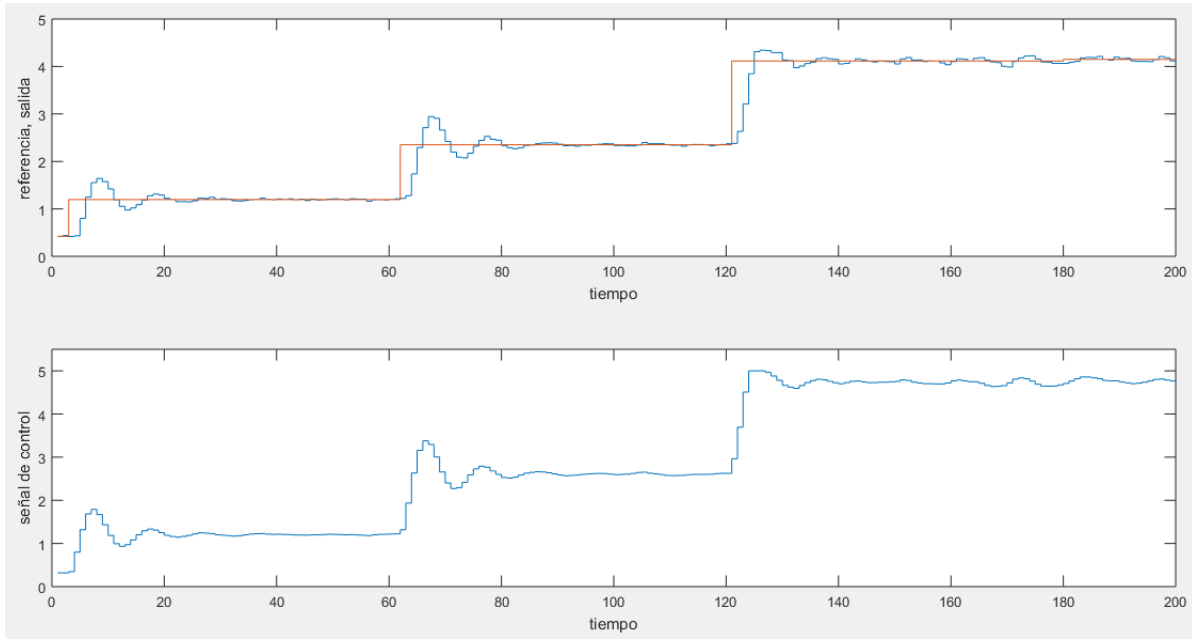


Figura 3.9 Inicio de la prueba para el caso subamortiguado. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa.

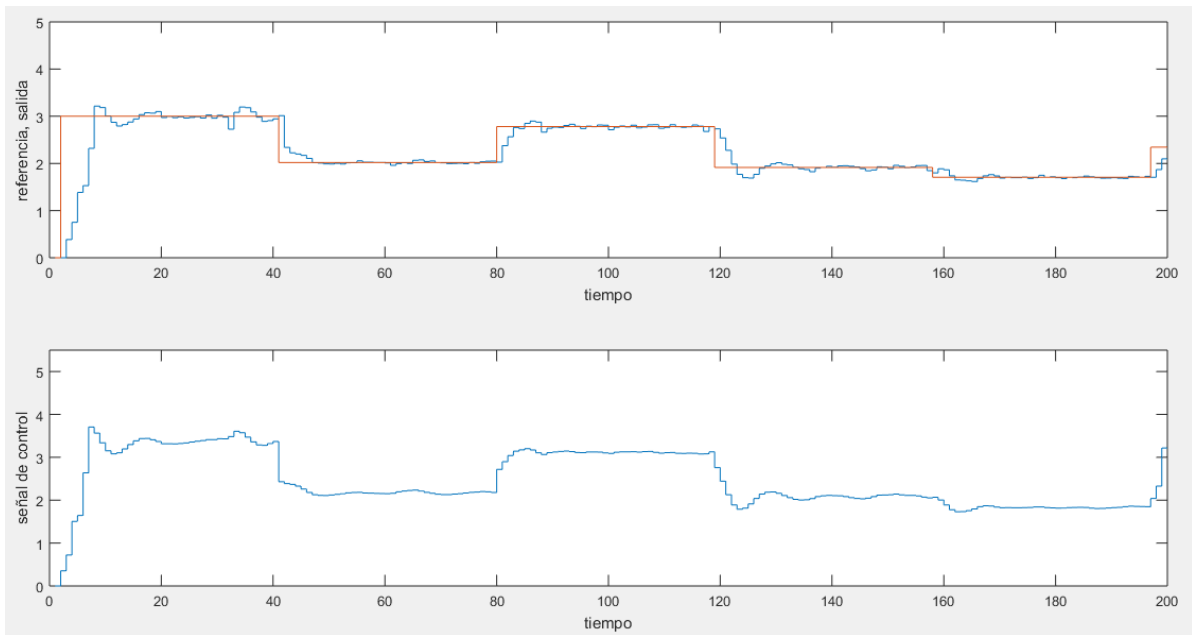


Figura 3.10 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo del cual la señal de salida alcanza la señal de referencia

e\de	NB	NM	NS	ZE	PS	PM	PB
NB	-2.3392	-2.08800002	-2.146401646	-2.525401646	-2.067801631	-1.0093	-0.0478
NM	-0.568512105	-0.36267269	-0.20674745	-0.848252855	0.007345141	-0.000500237	0.368002884
NS	-0.376520259	-5.62041036	-12.13032021	-6.688370341	-0.350860111	-0.047476619	1.564609979
ZE	-0.462720259	-2.35673148	-4.093104635	1.833024868	4.027886054	3.034352234	2.517964039
PS	-0.134300548	0.001065234	0.001836635	2.841622974	12.21007449	3.809315787	2.123964039
PM	-0.1338	0.103100103	0.170465598	1.712846964	0.582602541	2.18335872	2.026056168
PB	1.6964	2.1946	2.266331945	1.954031945	1.904984132	1.9571	8.0012

Tabla 5. Ajuste de los pesos (NFAM) después de un tiempo de iteración. Con variación de los rangos en 6.5424 y 28.6542 para NRME y NRMDE respectivamente y con el error de la diferencia de la señal de control deseada y la señal actual es de 0.0358

Después de haberse generado un entrenamiento adecuado al sistema neuro difuso se inicia los parámetros encontrados de los pesos, los rangos del error y de la derivada del error, y se aplica una perturbación al motor

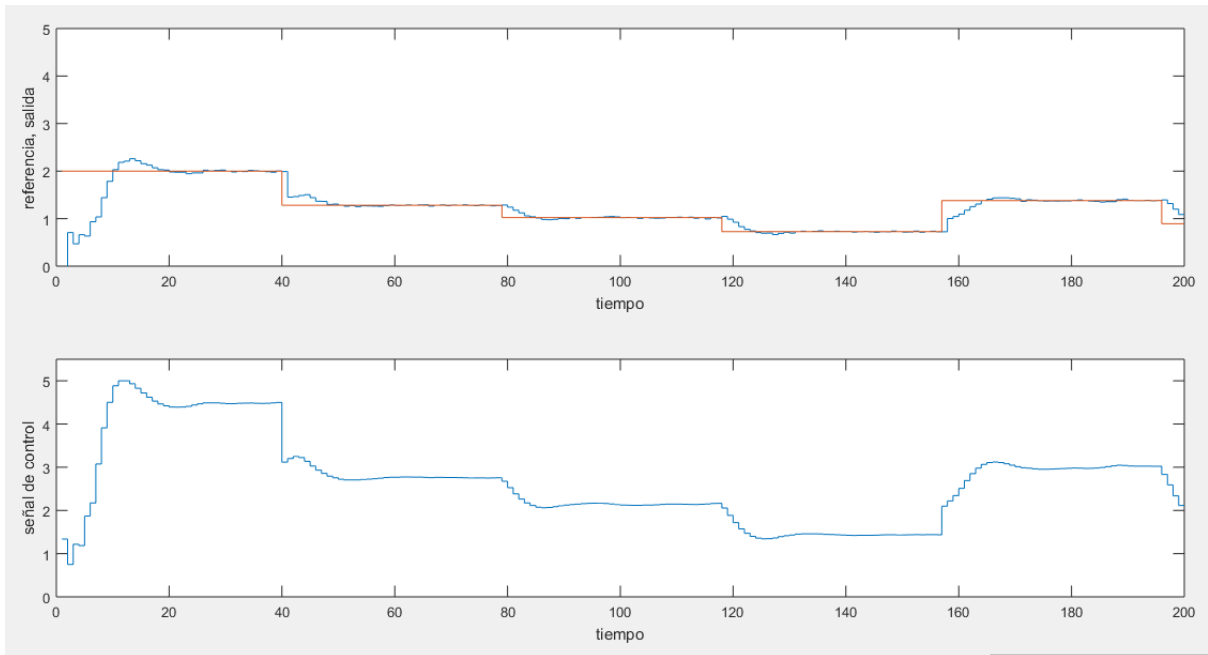


Figura 3.11 Comportamiento de la velocidad del motor bajo una perturbación

e/de	NB	NM	NS	ZE	PS	PM	PB
NB	-2.3392	-2.088	-2.1464	-2.5254	-2.0678	-1.0093	-0.0478
NM	-0.5683	-0.36230018	-0.142200521	-0.019400521	0.013899479	-0.000499999	0.7138
NS	-0.155	-0.89059334	-10.46614355	-5.897707261	0.006977026	-0.000299833	0.940700114
ZE	-0.2412	-1.70953667	-2.548397595	-0.618970939	8.701267487	1.48984548	1.862651637
PS	-0.1343	0.002938593	1.411697924	2.476936015	10.52907627	5.589332288	4.188200353
PM	-0.1338	0.104206775	0.736990236	2.909055742	0.582947254	2.208785245	2.051585202
PB	1.6964	2.194602826	2.266885519	1.954585519	1.905082703	1.9571	8.0012

Tabla 6. Ajuste de los pesos (NFAM) cuando la velocidad está bajo una perturbación. Con variación de los rangos en 4.4652 y 18.5634 para NRME y NRMDE respectivamente y con el error (E) de la diferencia de la señal de control deseada y la señal actual es de 0.0053

3.6.5. Caso oscilatorio

En este caso se establece las constantes K_p y K_i en 1 y 5 respectivamente, al inicio de la simulación como se observa en la figura 3.12 el comportamiento de la señal de control, de la forma oscilatoria. Después de un tiempo de entrenamiento se observa en la figura 3.13 el cambio de la señal de control; así mismo el funcionamiento de la planta bajo una perturbación como se observa en la figura 3.14.

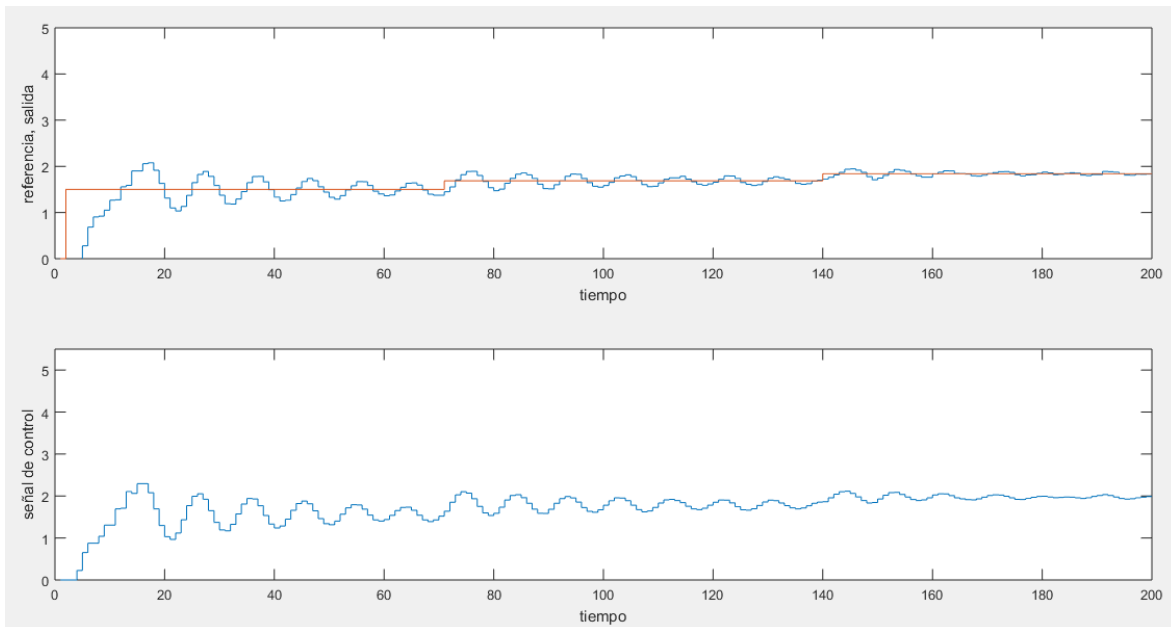


Figura 3.12 Inicio de la prueba para el caso oscilatorio. La primera grafica muestra la señal de referencia con la señal de salida de la planta y segunda grafica muestra la señal de control que sale de la red neuro difusa.

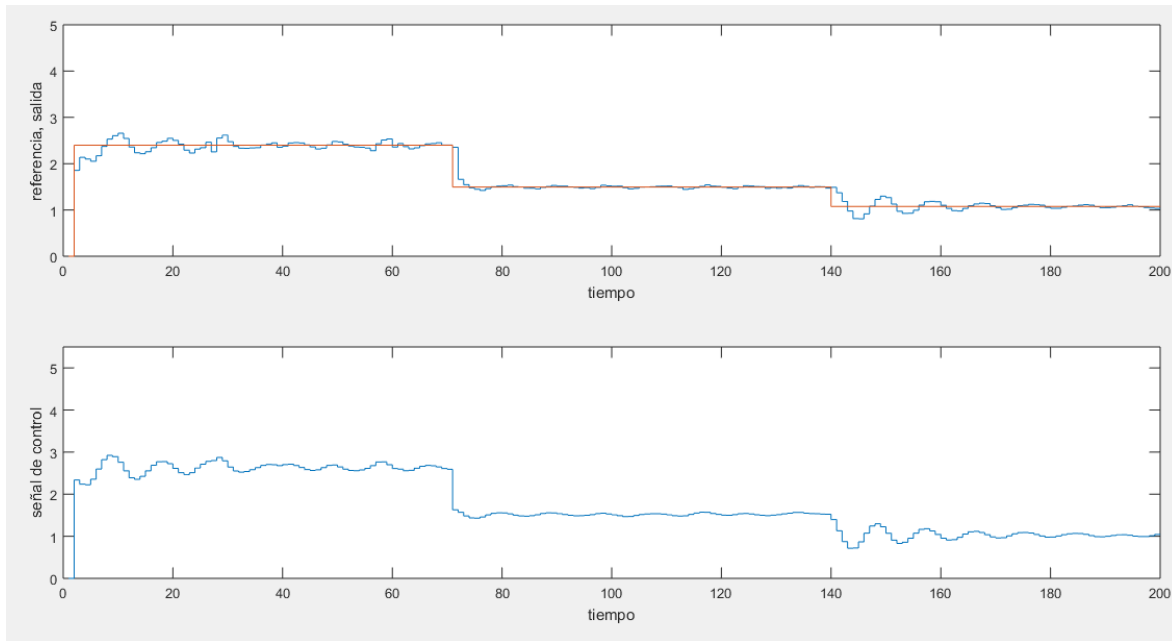


Figura 3.13 Respuesta de la señal de control después del ajuste de los pesos, reduciendo el tiempo del cual la señal de salida alcanza la señal de referencia

e\de	NB	NM	NS	ZE	PS	PM	PB
NB	-4.662258971	-0.48378818	-1.466789676	-2.246718752	-0.591054412	-0.393912635	-0.201404027
NM	-4.541225188	-0.01407668	-1.042213853	-2.190304945	0.010430941	7.5712E-05	0.017297248
NS	-1.505830289	-1.6532654	-9.602364153	-7.426165207	-0.009765194	-0.827663034	-0.041645329
ZE	-2.025510667	-3.22784755	-5.119385628	-0.233661468	5.20708305	3.138412168	0.189984076
PS	-1.770324488	-0.07535633	0.100846929	3.076884612	13.87289573	2.79897418	0.031876426
PM	-1.91690689	-0.09629253	0.284877648	1.083033061	7.106547055	1.589838392	0.017371219
PB	1.078894911	1.712069908	2.607740477	1.64353023	8.011247811	11.89214357	12.83533646

Tabla 7. Ajuste de los pesos (NFAM) después de un tiempo de iteración. Con variación de los rangos en 33.4742 y 58.6930 para NRME y NRMDE respectivamente y con el error de la diferencia de la señal de control deseada y la señal actual es de 0.4432.

Después de haberse generado un entrenamiento adecuado al sistema neuro difuso se inicia los parámetros encontrados de los pesos, los rangos del error y de la derivada del error, y se aplica una perturbación al motor

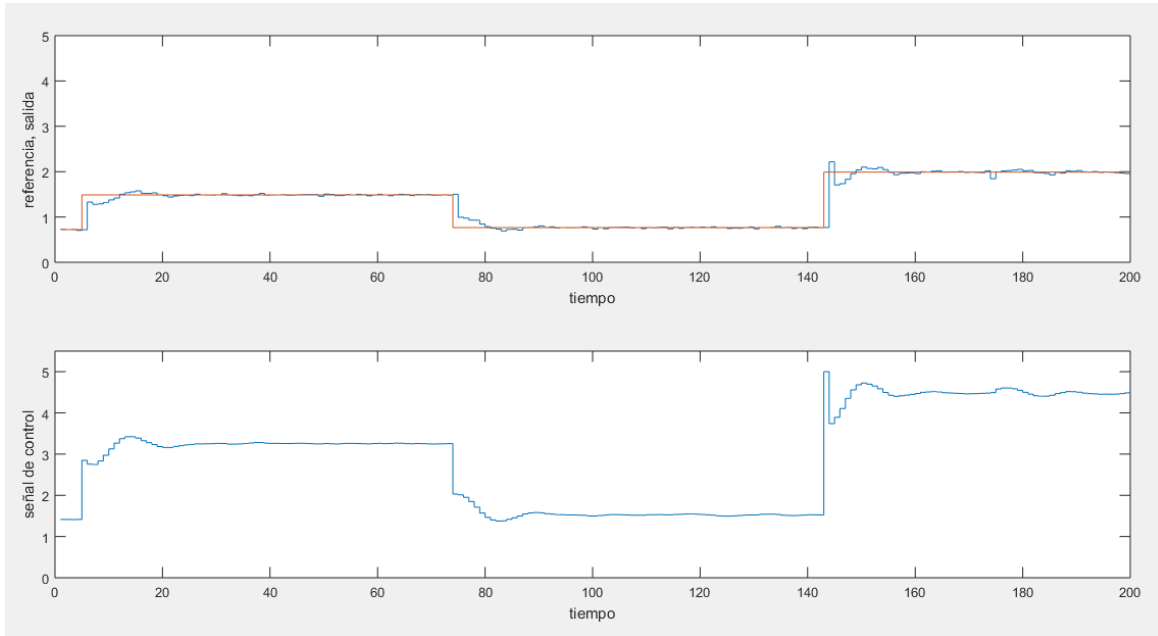


Figura 3.14 Comportamiento de la velocidad del motor bajo una perturbación

e/de	NB	NM	NS	ZE	PS	PM	PB
NB	-4.662258971	-0.48378818	-1.466789676	-2.246718752	-0.591054412	-0.393912635	-0.201404027
NM	-4.541225188	-0.01407668	-1.042213853	-2.190304945	0.010430941	7.5712E-05	0.017297248
NS	-1.505830289	-1.6532654	-9.602364153	-7.426165207	-0.009765194	-0.827663034	-0.041645329
ZE	-2.025510667	-3.22784755	-5.119385628	-0.233661468	5.20708305	3.138412168	0.189984076
PS	-1.770324488	-0.07535633	0.100846929	3.076884612	13.87289573	2.79897418	0.031876426
PM	-1.91690689	-0.09629253	0.284877648	1.083033061	7.106547055	1.589838392	0.017371219
PB	1.078894911	1.712069908	2.607740477	1.64353023	8.011247811	11.89214357	12.83533646

Tabla 8. Ajuste de los pesos (NFAM) cuando la velocidad está bajo una perturbación. Con variación de los rangos en 25.6372 y 43.7893 para NRME y NRMDE respectivamente y con el error (E) de la diferencia de la señal de control deseada y la señal actual es de 0.072.

CONCLUSIONES

4. CONCLUSIONES

- La correcta operación del sistema de lógica difusa depende únicamente de que tan minucioso sea el análisis del comportamiento de la planta y la descripción mediante sentencias lingüísticas obtenidas del operador, logrando así, evitar la difícil tarea de representar matemáticamente el sistema que se desea controlar.
- Las Redes Neuronales Artificiales resultan ser una herramienta muy útil para modelar sistemas no lineales. Sin embargo, la salida que se obtiene después de completar el proceso de aprendizaje depende de la constitución interna de la red y de las características del patrón de entrenamiento.
- El diseño de un controlador global depende de la naturaleza de la planta y generalmente suele ser complicado implementar una única técnica que posea la suficiente capacidad para arrojar los resultados esperados. Es por esto que resulta más sencillo emplear una combinación de controladores que actúen teniendo en cuenta los requerimientos de la planta.
- El entrenamiento del Sistema Neuro Difuso depende de qué valor sean las constantes iniciales de los rangos del error y de la derivada del error (NRME y NRMDE); aun así se deben limitar estos valores dependiendo de las constantes del entrenador PI (K_p y K_i), para que el entrenamiento de la red aprenda en ciertos rangos del error y de la derivada del error.
- La constante de aprendizaje (η) forma una parte importante del sistema, ya que dependiendo del valor de este, es la rapidez con que la red puede entrenarse, ya que, si son valores entre 0.5 y 1 la red tarda más en aprender, a diferencia de los valores por debajo de 0.5 la red aprende más rápido; aun así se establece un valor adecuado para el funcionamiento de la red.
- Una desventaja de este método radica en los altos costos de procesamiento que se deben realizar entre instantes de muestreo, pues se incorpora el tiempo de procesamiento tanto para el fusificador como para el defusificador y del entrenador PI al lazo de control, tiempo que es mayor en comparación al tiempo de procesamiento necesario.

- El sistema de control Neuro Difuso demuestra ser robusto ante perturbaciones introducidas en la señal de control logrando mantener el sistema cerca a la referencia. Además al considerar señales de referencia se pudo apreciar la sensibilidad de la planta ya que sigue este comando sin mayor dificultad.

BIBLIOGRAFIA

- [1] M. I. ACOSTA and C. A. ZULUAGA, Tutorial Sobre Redes Neuronales Aplicadas en Ingeniería Eléctrica. Universidad Tecnológica de Pereira, 2000.
- [2] M. T. HAGAN, H. B. DEMUTH, and M. BEALE, Neural Network Design. PWS Publishing Company, 1995.
- [3] L. ZADEH, "Fuzzy sets," *Information and control*, pp. 338-353, 1965.
- [4] "Is there need for fuzzy logic?" *IEEE*, pp. 1-3, 2008.
- [5] S. KOURO and R. MUSALEM, "Control mediante logica difusa," *Tecnicas modernas automaticas*, pp. 1-7, 2002.
- [6] A. ALZATE, Control Difuso. Universidad Tecnologica de Pereira, 2001.
- [7] J. HILERA and V. JIMENEZ, Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones., Rama, Ed., 1995.
- [8] D. GIRALDO and I. TABARES, "Control de velocidad de un motor de utilizando logica difusa," *Scientia Et Technica*, no. 9, pp. 111-119, Abril 1999.
- [9] H. ACOSTA, "Diseño de controladores dedicados a la lógica difusa." Ph.D. dissertation, Universidad Autónoma de Madrid, Octubre 2006.
- [10] Zhang, Y. Q., & Kandel, A. (1998). Compensatory neurofuzzy systems with fast learning algorithms. *Neural Networks, IEEE Transactions on*, 9(1), 83-105.
- [11] Lin, C. T., Juang, C. F., & Li, C. P. (1999). Temperature control with a neural fuzzy inference network. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 29(3), 440-451.
- [12] Obregón, R. F., & Arroyave Bustos, L. A. (2012). *Herramienta para el diseño y simulación de sistemas neuro-difusos en matlab NF-Tool [recurso electrónico]* (Doctoral dissertation).

[13] Caballero Suárez, J. A., Ruiz Femenia, J. R., & Aracil Sáez, I. (2011). Optimización (curso 2011-2012). *Simulación y Optimización de Procesos Químicos*.

[14] Hilera, José R. y Martínez, Víctor J. Redes neuronales artificiales (Fundamentos, modelos y aplicaciones). Madrid, España: Alfaomega Ra-Ma, 2000.

[15] L. E. M., CHAVEZ, N. P., & MACÍAS, M. V. B. APRENDIZAJE NO SUPERVISADO Y EL ALGORITMO WAKE-SLEEP EN REDES NEURONALES. UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA, 2012