

CONTROL DIFUSO DE TEMPERATURA PARA CARGA RESISTIVA CON MICROCONTROLADOR DE 8 BITS

Ing. GUSTAVO ADOLFO MENESES BENAVIDES

ESPECIALIZACION EN CIENCIAS ELECTRONICAS E INFORMATICA AUTOMATIZACION
INDUSTRIAL

Director: Ing. AMADO TAVERA CRESPO

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERIA
DEPARTAMENTO DE ELECTRÓNICA

Medellín; 2005

Tabla de contenidos

PROLOGO . .	1
1. INTRODUCCION .	3
2. ESTRUCTURA DE UN CONTROLADOR DIFUSO . .	5
2.1 BASE DE CONOCIMIENTOS .	6
2.2 MOTOR DE INFERENCIAS .	6
2.3 FUZIFICADOR .	7
2.4 DEFUZIFICADOR . .	8
3. DISEÑO DE CONTROLADORES DIFUSOS . .	11
3.1 SELECCION DE LAS VARIABLES Y DEL UNIVERSO DE DISCURSO .	12
3.2 SELECCIÓN DE LA ESTRATEGIA DE FUZIFICACION . .	13
3.3 CONSTRUCCION DE LA BASE DE REGLAS .	13
3.4 SELECCIÓN DE LA LOGICA DE TOMA DE DECISIONES .	15
3.5 SELECCIÓN DE LA ESTRATEGIA DE DEFUZICACION . .	15
4. MICROCONTROLADORES DE 8 BITS .	17
4.1 EL MICROCONTROLADOR PIC16F876 . .	18
4.2 PROGRAMACION DE MICROCONTROLADORES DE 8 BITS .	24
4.3 REFERENCIA DE DECLARACIONES UTILIZADAS: . .	26
4.3.1 OPERADORES MATEMÁTICOS .	27
4.3.2 OPERADORES DE COMPARACION .	28
4.3.3 OPERADORES LOGICOS . .	29
4.4 OTRAS DECLARACIONES UTILIZADAS . .	29
4.5 CONVERSION ANALOGA/DIGITAL .	30
4.5.1 CONFIGURACION DE REGISTROS DEL MODULO A/D . .	31
4.6 SENSOR DE TEMPERATURA .	33
5. MEMORIAS DE DISEÑO . .	35
5.1 MODELADO DIFUSO DEL SISTEMA .	35

5.1.1 FUNCIONES DE PERTENENCIA Y UNIVERSOS DE DISCURSO . .	36
5.2 ESTRATEGIA DE CONTROL .	43
5.3 IMPLEMENTACION DEL CONTROLADOR DIFUSO . .	44
5.3.1 CONJUNTO DE REGLAS DEL CONTROLADOR .	44
5.3.2 INFERENCIA DE LA ACCION DE CONTROL .	45
5.3.3 ALGORITMO DE CONTROL . .	48
5.3.4 PROGRAMA PARA EL MICROCONTROLADOR .	49
5.3.5 PLANO ELECTRICO DEL MONTAJE . .	66
6. RESULTADOS . .	69
7. CONCLUSIONES . .	73
BIBLIOGRAFIA .	75

PROLOGO

El objetivo de este trabajo es mostrar en detalle el proceso de implementación del núcleo principal de un sistema de control difuso de temperatura sobre un dispositivo de la electrónica programable moderna, como es un microcontrolador de 8 bits. En este informe se presentan las bases teóricas generales de los controladores difusos, a continuación se presenta una descripción completa de los elementos utilizados en el sistema de control y luego se presenta el proceso de construcción sobre el microcontrolador de los bloques de fuzificación, defuzificación e inferencia basada en reglas, junto con los resultados obtenidos.

Teniendo en cuenta que, hasta ahora, en el ámbito local se ha producido escasa literatura en el área de controladores difusos, este trabajo es importante ya que constituye un aporte para la construcción de una base de conocimiento más amplia en este campo. A través del informe se entregan guías y referencias puntuales sobre el proceso práctico de construcción de controladores difusos las cuales pueden resultar de utilidad para personas pertenecientes a diferentes entornos académicos o productivos que posean conocimientos sobre el tema y deseen iniciarse o profundizar en el campo de la experimentación asociada con el control difuso.

Este trabajo está enmarcado dentro del área de control, específicamente fue motivado y además esta sustentado en los conocimientos adquiridos en las asignaturas: Control Automático, Instrumentación y Control Inteligente, cursadas en la Especialización de Ciencias Electrónicas e Informática, área de Automatización Industrial, de la Universidad de Antioquia, en Medellín Colombia.

1. INTRODUCCION

Presentado en 1974 como una tecnología emergente que apuntaba hacia las aplicaciones industriales, el control difuso le ha agregado una dimensión prometedora al dominio existente en la ingeniería de control convencional. Cuando un sistema complejo físicamente no entrega un modelo matemático preciso o razonablemente aproximado y en particular cuando la descripción del sistema requiere de cierta experiencia, en términos generales, la metodología del control difuso tiene características sobresalientes y meritos distinguidos sobre muchas otras técnicas. La utilización del conocimiento humano en el diseño de controladores no solo representa una ventaja sino que es a menudo necesario. La relativamente nueva tecnología de control difuso tiende a ser una alternativa más que un reemplazo de las técnicas de control existentes.

Comparada con los enfoques convencionales, el control difuso utiliza más información de los expertos del área, acoge más un cierto tipo de heurística y se apoya menos en la modelación matemática de un sistema físico. El proceso de escogencia de un conjunto difuso o de una base de reglas difusa puede parecer bastante subjetivo si se trata de tener en cuenta el conocimiento humano sobre los sistemas físicos internos de una planta. Sin embargo esta es la misma situación que escoger un modelo en control determinístico o una distribución en control estocástico.

La teoría de control difuso puede ser rigurosa y los controladores de este tipo pueden tener estructuras precisas y analíticas con estabilidad de lazo cerrado garantizada si se desea tener tales características. La tecnología de control difuso se está transformando como parte de la teoría de control moderna. La tendencia de un enfoque riguroso al

control difuso, la cual empezó desde mediados de los 80, ha producido muchos resultados excitantes y promisorios. Por ejemplo, algunas estructuras analíticas de controladores difusos y su relación con los controladores convencionales correspondientes son mucho mejor comprendidas hoy. Se han desarrollado numerosos métodos de análisis y diseño los cuales han transformado la antes llamada “arte” de construir un controlador difuso funcional en la “ciencia” del diseño sistemático. Otro aspecto importante es que la teoría de control analítico ha hecho la práctica del control difuso más segura, más eficiente y más costo-efectiva. [14] [2]

En la década del 90 el control difuso ganó una gran aceptación en la industria y la academia. El éxito mundial de innumerables productos comerciales y aplicaciones ha probado que la tecnología difusa no solo es práctica y poderosa sino también costo-efectiva. Los sistemas del mundo real son no lineales; la modelación exacta es difícil, costosa e incluso imposible en muchos casos. El control difuso tiene la habilidad única para realizar exitosamente tareas de control sin conocer el modelo del sistema aun si este es no lineal y complejo. [15]

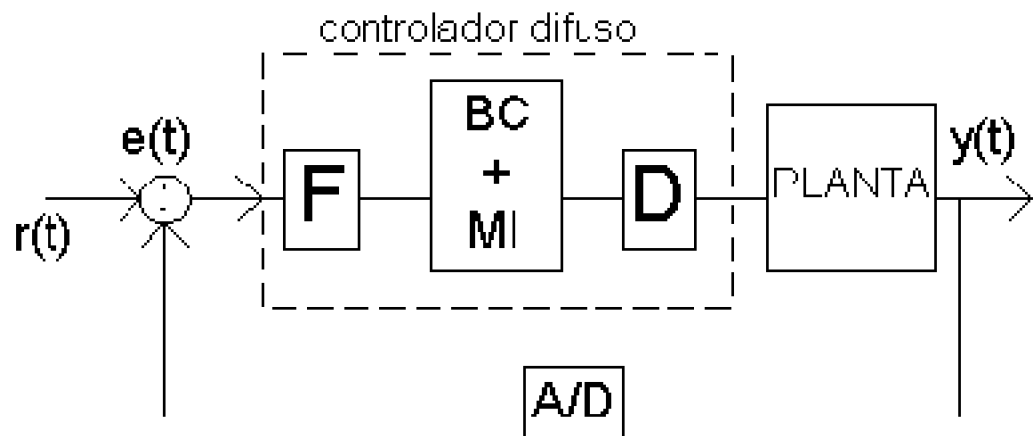
En hardware los sistemas difusos dominan las tendencias actuales en aplicaciones con microprocesadores y chips VLSI con diseños específicos. [16]

En la actualidad un campo importante en la investigación es el diseño de hardware difuso mejorado. Investigaciones recientes en la literatura científica han mostrado el interés creciente en las implementaciones análogas y digitales de controladores difusos en particular y de sistemas difusos en general. Desarrollos VLSI y análogos de sistemas difusos en la forma de arquitecturas dedicadas apuntan a la mas alta eficiencia en implementación. Debido a eso día a día se trabaja para aumentar la velocidad de procesamiento y mejorar la utilización del silicio en los chips que se utilizan en este tipo de aplicaciones. [17]

2. ESTRUCTURA DE UN CONTROLADOR DIFUSO

Un Controlador basado en lógica difusa es un sistema electrónico e informático que sirve de soporte físico para realizar inferencias con lógica difusa. Este dispositivo funcionará en los niveles inferiores de la pirámide de control, cerrando lazos de control en forma similar al PID convencional, aunque con ventajas adicionales respecto de éste. Fundamentalmente porque podrá aplicarse con sencillez a procesos complejos y pobremente definidos ya sea porque no se dispone de conocimiento cuantitativo de su dinámica o porque la información proveniente de sensores es imperfecta (faltante, imprecisa, corrupta por ruido, etc.).

La arquitectura general de un controlador difuso monolazo, para una sola entrada y una sola salida, se aprecia en la figura 2.1. Se considera que la variable controlada es de tipo analógica y que debe ser convertida a digital antes de ser usada en los cálculos internos del controlador. Por lo tanto, el controlador difuso será un caso particular de un controlador digital, cuyo comportamiento será no lineal. Sus bloques constitutivos son una Base de Conocimiento (BC), una Base de Datos (BD) o memoria de trabajo, un Motor de Inferencias (MI), un Fuzificador (F) y un Defuzificador (D).



Arquitectura General de un Controlador Difuso

2.1 BASE DE CONOCIMIENTOS

La arquitectura que se aprecia en la figura para el controlador difuso es la de un sistema basado en conocimiento. Para representar el conocimiento existen muchos métodos, aunque para los sistemas de tipo reactivos (que responden a las

entradas provenientes de sensores) como los controladores difusos el método preferido suele ser la representación en forma de reglas. Generalmente m reglas del tipo:

Si X_1 es A_1 y X_2 es A_2 y.....y X_n es A_n entonces Y es B

Formula

Donde las X_i son salidas del sistema a controlar, o funciones de dicha salida (errores, derivadas), provenientes de sensores o estimadores; las Y son directamente las variables manipuladas que tendrán que ser convertidos a valores únicos para manejar los actuadores; y los A_i son conjuntos difusos. Mediante reglas de este tipo se codifica en el sistema informático parte del conocimiento humano en el manejo del proceso, así como también los objetivos de control. Por ejemplo: "Si error de temperatura es Grande Positivo y la derivada error de temperatura es Grande Positiva entonces aumento de combustible inyectado es Grande Negativo"

2.2 MOTOR DE INFERENCIAS

Es el encargado de concatenar las implicancias de la Base de Conocimiento, aplicando recursivamente una regla de inferencia composicional. Por ejemplo, el operador max/min. Básicamente un proceso de inferencia difusa incluye dos conceptos; la implicación difusa y la regla composicional de inferencia. Si tenemos las observaciones A' y B' , entonces la

conclusión difusa C' puede ser inferida por

$$C' = (A' \circ B') \circ R$$

Formula

Donde “o” representa el operador composicional. Dentro de las numerosas estrategias de inferencia, la técnica de Mamdani es la más utilizada en los sistemas difusos de control existentes debido a su simplicidad. En este método, la operación del mínimo se adopta para calcular una relación de implicación difusa y el max-min como la regla composicional de inferencia. La conclusión deducida de la regla i -ésima C'_i para las entradas A' y B' se puede calcular con la regla composicional max-min de inferencia como sigue:

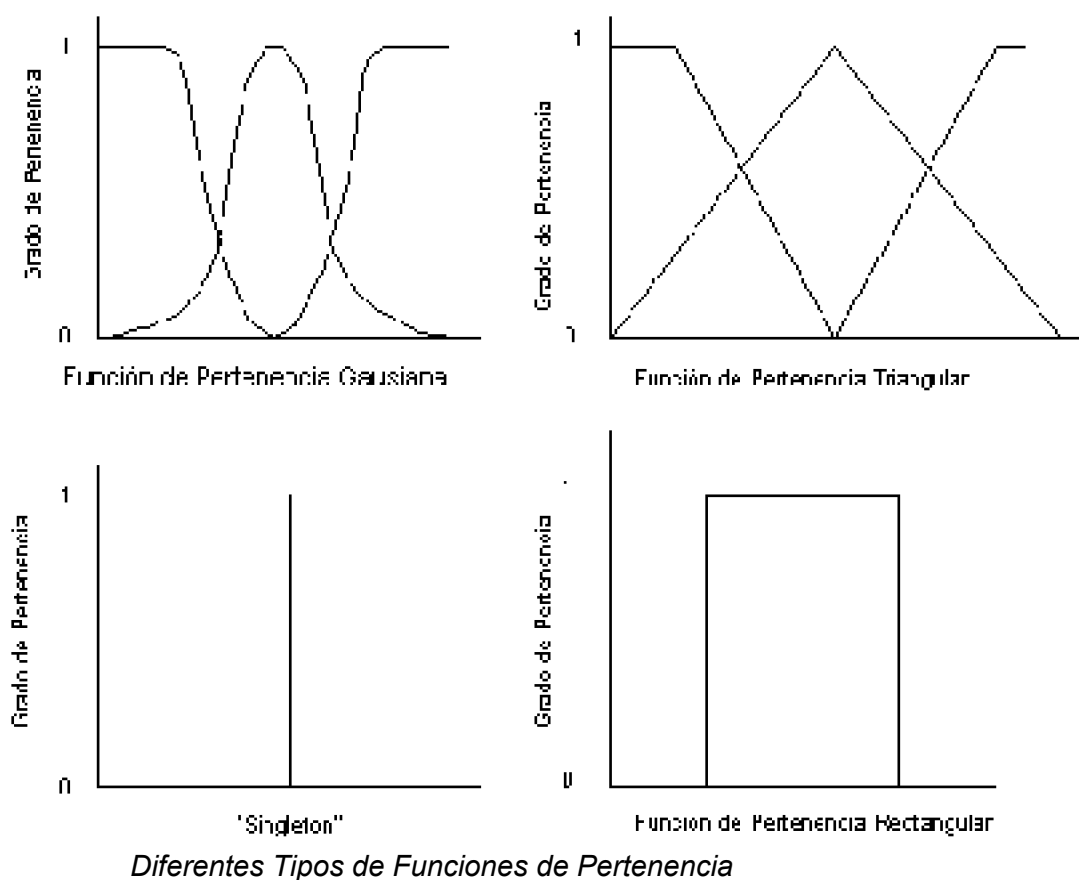
$$\mu_{C'_i}(z) = \max_{x_1 \in U} \max_{x_2 \in V} \{ [\mu_{A'}(x_1) \mu_{B'}(x_2)] \mu_{R_i}(x_1, x_2, z) \}$$

Formula

2.3 FUZIFICADOR

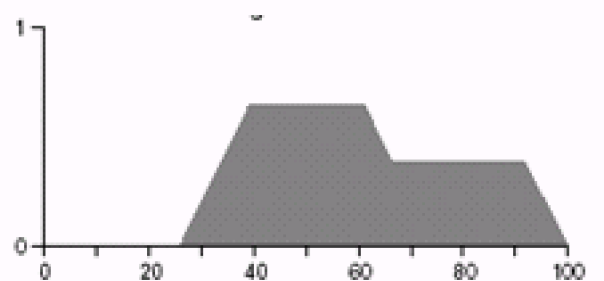
La función de este bloque es la de asociar a las entradas una función de pertenencia, acorde con la partición del universo de discurso que se haya efectuado. El fuzificador transforma la medición del valor determinístico de una variable en valores subjetivos mediante un proceso de verbalización. De esta manera, se pueden distinguir las siguientes funciones:

- medir los valores de las variables de entrada;
- hacer una adaptación de escalas desde estas variables de entrada y el universo de discurso, en una suerte de normalización;
- asociar a la entrada un valor lingüístico que puede verse como la pertenencia a un dado conjunto difuso.



2.4 DEFUZIFICADOR

A partir de los consecuentes de las reglas que se hayan disparado, es necesario obtener una única acción de control, combinando el resultado propuesto por cada una de ellas. Para ello, este bloque es el encargado de servir de soporte para la aplicación de alguna de las técnicas elegidas (el máximo, la media de máximos, el centroide).



Resultado de un Proceso de Inferencia Difusa

La defuzificación se realiza usualmente mediante cálculos promedio. Uno de los métodos mas conocidos es el del Centro de Gravedad (COG), el cual es utilizado para calcular el baricentro de una masa. Una representación discreta del COG es

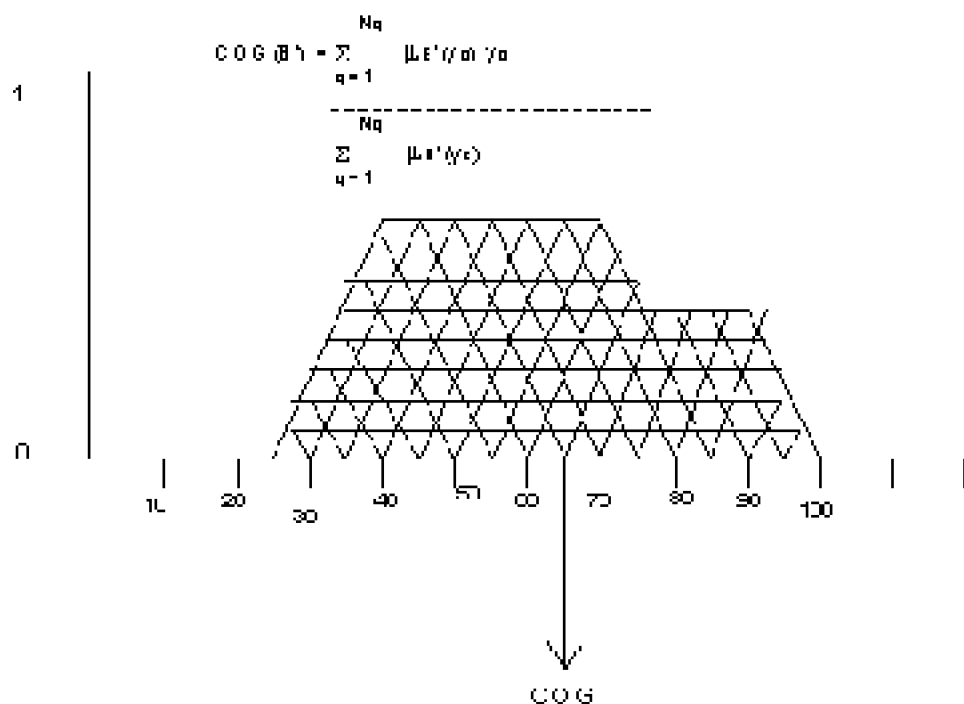
$$N_q$$

$$\text{COG}(B') = \sum_{q=1}^{N_q} \mu_{B'}(y_q) y_q$$

$$N_q$$

$$\sum_{q=1}^{N_q} \mu_{B'}(y_q)$$

Formula



Ubicación del COG

donde N_q es el numero de pasos de cuantización por los cuales el universo de discurso Y de la función de pertenencia $\mu_{B'}(y)$ es discretizada. La grafica muestra la ubicación del COG para el conjunto difuso de salida. Además de los métodos clásicos, basados en el concepto de conjuntos difusos de salida, existen diversos métodos de defuzzificación computacionalmente simplificados mediante los cuales es posible combinar la agregación y defuzzificación en una sola fase. Uno de ellos es el Promedio de los Pesos (WM):

R

$$WM(B') = \sum_{r=1}^R \theta_r b_r$$

R

$$\sum_{r=1}^R \theta_r$$

Formula

Donde R es el numero de reglas, θ_r es el grado de activación de la regla r-ésima y b_r es un valor numérico asociado con el consecuente B' de la regla.

3. DISEÑO DE CONTROLADORES DIFUSOS

Aunque no existe un procedimiento estándar, es posible definir ciertos pasos de trabajo para llegar a un buen diseño del controlador difuso. La metodología que se propone sigue los lineamientos generales expuestos por LEE [18] aunque con algunas modificaciones. [13]. Es un hecho que no existe una solución única, y que el diseño se asienta en conocimiento heurístico sobre el proceso y pruebas de ensayo-error hasta conseguir una respuesta adecuada a las especificaciones previas, en un proceso de refinamiento que la mayoría de las veces requiere una nueva definición en el criterio de selección usado en estos pasos.

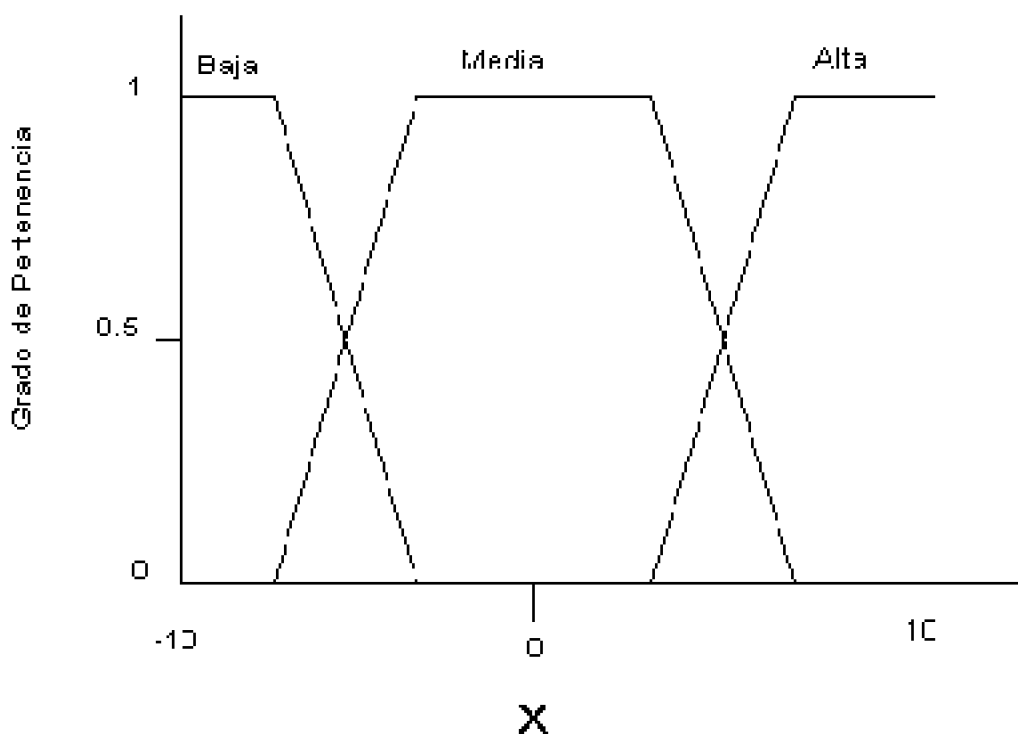
La “sintonización” del controlador puede hacerse también en forma automática, por ejemplo con algoritmos genéticos, o redes neuronales, de modo tal de encontrar un controlador óptimo. Incluso esta optimización puede hacerse en línea, una vez que el controlador está en funcionamiento normal en planta, aunque deben contemplarse especialmente los aspectos de estabilidad. Sin embargo, este tipo especial de controlador difuso denominados adaptivos, es objeto de otro campo de trabajo. Una metodología propuesta se resume en los siguientes pasos:

1. Selección de las variables y del Universo de Discurso
2. Selección de la estrategia de fuzificación
3. Construcción de la base de reglas

4. Selección de lógica de toma de decisiones
5. Selección de la estrategia de defuzificación

3.1 SELECCION DE LAS VARIABLES Y DEL UNIVERSO DE DISCURSO

Como en todo lazo de control, el primer paso en el diseño es especificar claramente cuáles serán las variables de entrada y salida que queremos relacionar a través del controlador. Además es necesario conocer de antemano su rango de variación. Para esto se pueden presentar dos situaciones bien diferentes. El controlador difuso puede desarrollarse con base en el conocimiento de un experto humano, en cuyo caso esta etapa se completa rápidamente a partir de la experiencia previa. Si en cambio es la primera aproximación que tenemos al problema, habrá que elegir las variables que incidan directamente sobre la variable controlada, trabajando generalmente sobre el error o alguna función del error. Estas serán las variables provenientes de los sensores. Luego es necesario hacer sencillas maniobras de identificación de rangos. Por ejemplo desde una posición de equilibrio como el reposo, puede efectuarse una excursión máxima de la variable manipulada y volverla a cero, para que el proceso evolucione en forma libre a su posición de equilibrio nuevamente. En esta situación puede medirse la salida y tenerse una noción de la amplitud del error y su variación (construcción de un plano de fase). [13]



Ejemplo de Funciones de Pertenencia a lo largo de un Universo de Discurso

3.2 SELECCIÓN DE LA ESTRATEGIA DE FUZIFICACION

En esta etapa se determina en qué grado las variables de entrada pertenecen a los conjuntos difusos a través de su función de pertenencia. Se trata de desarrollar un bloque que tendrá en sus entradas los valores numéricos elegidos en la etapa anterior, y a su salida dichos valores asociados a conjuntos difusos con sus correspondientes funciones de pertenencia. Para ello es necesario definir estos conjuntos o universo de discurso, y luego su soporte, en un procedimiento que se conoce como *partición* del universo de discurso, y obviamente ya requiere de un conocimiento a priori sobre el proceso a controlar.

La cantidad de conjuntos difusos de nuestro universo de discurso es también una decisión de diseño. Cuanto más cantidad elijamos, mayor poder descriptivo y flexibilidad tendrá nuestro lenguaje, pues no olvidemos que estos conjuntos difusos son directamente las etiquetas lingüísticas con que nos referiremos a las variables de entrada y de salida. En el caso de trabajar con el error, valores típicos son positivo pequeño (PP), negativo mediano (NM), y otros. Obviamente, este mayor poder descriptivo tiene su costo en mayor complejidad por la cantidad de reglas involucradas, que crece proporcionalmente. En la práctica, se suelen elegir entre 2 y 10 etiquetas lingüísticas para cada variable de entrada o salida, luego de un procedimiento de prueba y error.

Respecto de la *forma* que tendrán las funciones de pertenencia, no existen diferencias apreciables en los resultados de control entre las diferentes posibilidades (triangulares, trapezoidales, gaussianas, sigmoideas, etc.). Por ello la mayoría de los productos comerciales suelen ofrecer las trapezoidales, entendiendo la triangular como un caso particular, ya que son las de más fácil implantación.

Si bien hasta el momento toda referencia a la partición de los universos de discurso fue hecha con mayor énfasis en el de las entradas, su validez se extiende a las salidas del controlador difuso. La única salvedad importante es que las salidas del controlador serán siempre números reales, ya que a partir de estos se generarán los valores de tensión, corriente u otra magnitud física que manejen los actuadores del sistema de control bajo diseño. Se espera que el controlador provea una salida para todos los estados posibles de la planta a controlar. Para ello, todas las variables de entrada y salida del controlador difuso deben poder vincularse a un conjunto difuso, y a su vez cada conjunto difuso debe participar del antecedente o consecuente de una regla.

3.3 CONSTRUCCION DE LA BASE DE REGLAS

Ya se vio que el conocimiento del controlador se almacenaba en forma de reglas del tipo "if...then" que se combinaban entre si para formar un único conjunto difuso de salida. Existen diferentes alternativas para obtener dichas reglas. Este es el punto más importante de la programación del controlador difuso.

Basadas en la experiencia y el conocimiento de un ingeniero de control experto en el proceso a controlar. Esta es quizá la manera más sencilla de obtener el conocimiento necesario para resolver el problema, y es la situación para la cual fueron ideados en primera instancia estos controladores. Este conocimiento está organizado en el experto en forma de variables que se describen con el lenguaje natural. De este modo, los términos de este lenguaje natural se traducen a etiquetas lingüísticas y se establecen las relaciones entre los diferentes conceptos, que quedarán plasmadas en las reglas. El procedimiento aplicable en este caso puede ser bien un relato introspectivo del propio experto, como suele presentarse en los manuales de operación de los hornos de cemento, o bien un interrogatorio guiado, hecho por alguien que también conozca sobre el proceso a controlar, que actuará como ingeniero de conocimiento. A partir de un cuerpo mínimo de reglas se implantará el primer prototipo, cuyo crecimiento será incremental, basado fundamentalmente en prueba y error hasta conseguir la respuesta deseada.

Basadas en la observación de un operario experto. Existen muchas situaciones de control en las que es muy difícil hallar un modelo por la cantidad y complejidad de variables involucradas. Ello hace impensable la aplicación de una técnica de control convencional. La idea central entonces en este modo de construir la base de conocimiento consiste en volcar en forma de reglas con variables lingüísticas la manera de llevar adelante el control.

Basadas en un modelo difuso del proceso. Es posible hacer una descripción lingüística no sólo de la estrategia de control, sino también de las características dinámicas del proceso a controlar. En términos generales, esto es un modelo difuso del proceso. A partir de este modelo se pueden obtener las reglas difusas para controlarlo, incluso optimizando alguna figura de mérito. Un modelo muy utilizado es el paramétrico, propuesto por el Prof. Sugeno, realizado con reglas del tipo:

$$R_i: \text{IF } s_1 \text{ is } S_1 \text{ AND } s_2 \text{ is } S_2 \text{ AND } s_3 \text{ is } S_3 \text{ AND } \dots s_p \text{ is } S_p$$

$$\text{THEN } v_i = a_0 + a_1 s_1 + a_2 s_2 + \dots + a_p s_p$$

Formula

donde las S_j son las etiquetas lingüísticas, s_i las entradas, v_i las salidas, y los a_j son los parámetros a identificar.

Basadas en aprendizaje. Su precursor fue el Profesor Mamdani, con el controlador auto-organizado, que está formado por dos cuerpos de reglas. Uno que hace el control propiamente dicho, y otro, a un nivel supervisor de aquel, que exhibe el comportamiento que seguiría un programador humano mejorando las reglas de control. A este tipo de reglas que actúan sobre las reglas de control se las suele denominar "meta-reglas". Notar

que si bien este tipo de controladores difusos mostrarán un comportamiento adaptivo, conviene distinguirlos de los ya presentados como tales, en tanto y en cuanto éstos se adaptan mediante nuevas reglas, mientras que aquellos lo hacen modificando reglas, aunque también las funciones de pertenencia de los conjuntos difusos, tanto de entradas como de salidas.

3.4 SELECCIÓN DE LA LOGICA DE TOMA DE DECISIONES

En concordancia con el tipo de reglas se procede a diseñar una matriz de inferencia que resume las acciones de control que se tomarán de acuerdo a las diversas condiciones posibles. El controlador debe generar salidas para cualquier estado difuso de entrada. En efecto, si para todo valor de las variables de entrada y salida existe un conjunto difuso al que pertenecen y una regla que involucra dichos conjuntos difusos, se satisfecerá que el controlador genere una salida para cualquier combinación de las entradas. En cambio si detectamos que este no es el caso, debe incorporarse una regla que cubra la condición no incluida. Trabajando con la matriz de inferencias, es difícil dejar fuera situaciones como no previstas.

En efecto, por simple inspección se puede apreciar si hay situaciones para las que no se da salida del controlador difuso. Del mismo modo esta condición se viola si alguna etiqueta lingüística no está presente o, lo que es lo mismo, no se ha descrito todo el soporte del universo de discurso. En un controlador difuso las reglas interactúan entre sí, vale decir, la presencia de una de ellas modifica el comportamiento del conjunto y viceversa. Este conjunto es el que forma la superficie de control final. Supóngase añadir la regla i -ésima a la base. Si esta superficie de control se representa con R y las entradas con X , la composición de ambas da un nuevo conjunto difuso Y , que es distinto para la nueva situación de lo que era previamente. Puede demostrarse [19] que la interacción de las reglas depende del tipo de implicancia que se elija en la materialización del controlador difuso.

Las reglas contradictorias, que propongan acciones de control opuestas, deben ser eliminadas o reemplazadas en la base de conocimiento. En un controlador difuso en el que no interviene una gran cantidad de reglas, la consistencia puede determinarse por simple inspección. Sin embargo, para bases con un mayor número de reglas es necesario hacer una revisión más sistemática. Existen métodos para ello basados en las definiciones de Posibilidad ($\mathcal{D}$) y Necesidad ($\mathcal{I}$), a partir de las que puede obtenerse un índice de inconsistencia para cada regla respecto del conjunto de la base [19].

3.5 SELECCIÓN DE LA ESTRATEGIA DE DEFUZICACION

Una vez cumplidas las etapas anteriores, el resultado de la agregación de las reglas de la base, es decir, la inferencia, será también un conjunto difuso. Sin embargo, para manejar el actuador es necesaria una magnitud no difusa. En efecto, si dicho actuador es una válvula automática, por ejemplo, el controlador no puede dar como salida la aplicación de una “pequeña tensión positiva” al servomotor que maneje la válvula, a menos que la válvula admita este tipo de entradas. De todos modos, si las admite, siempre estamos transfiriendo a otro elemento del sistema de control la solución del problema. Es por lo tanto necesario un bloque que sea capaz de dar una salida determinada (“*crisp value*”) frente a una entrada difusa, y que represente lo mejor posible la distribución de posibilidades de la acción de control inferida. Este es justamente el rol del defuzificador. Para desarrollar este bloque, las estrategias más comúnmente empleadas son los criterios del máximo, del promedio de máximos y del centro de área.



4. MICROCONTROLADORES DE 8 BITS

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de *controlador empotrado* (embedded controller).

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear

microcontroladores más potentes y consecuentemente más caros.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

Los microcontroladores más populares se encuentran, sin duda, entre las mejores elecciones:

8048 (Intel). Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que toda-vía sea popular.

8051 (Intel y otros). Es un microcontrolador muy popular. Fácil de programar y de gran capacidad. Está bien documentado y posee cientos de va-riantes así como numerosas herramientas de desarrollo.

80186, 80188 y 80386 EX (Intel). Son versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permi-ten aprovechar las herramientas de desarrollo para PC.

68HC11 (Motorola y Toshiba). Es un microcontrolador de 8 bits con gran capacidad de realizar tareas, tiene gran cantidad de variantes.

683xx (Motorola). Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altas presta-ciones.

PIC (Microchip). Familia de microcontroladores que ha ganado una gran popularidad. Fueron los primeros microcontroladores RISC. Tiene un conjunto reducido de instrucciones para programación a bajo nivel.

Es preciso resaltar en este punto que existen innumerables familias de mi-crocontroladores, cada una de las cuales posee un gran número de variantes.

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Los microcontroladores PIC responden a la arquitectura Harvard.

4.1 EL MICROCONTROLADOR PIC16F876

Este dispositivo pertenece a los microcontroladores de la gama media de la firma Microchip y presenta las siguientes características principales de funcionamiento:

Frecuencia de Operación: 0 a 20MHz (Usualmente se trabaja de 4MHz en adelante).

8K de Memoria de Programa

Memoria de datos RAM : 368 bytes

Memoria de datos EEPROM: 256 bytes

3 Puertos de Trabajo I/O (Puerto A, Puerto B y Puerto C)

5 entradas de Conversor Análogo/Digital de 8 o 10 bits

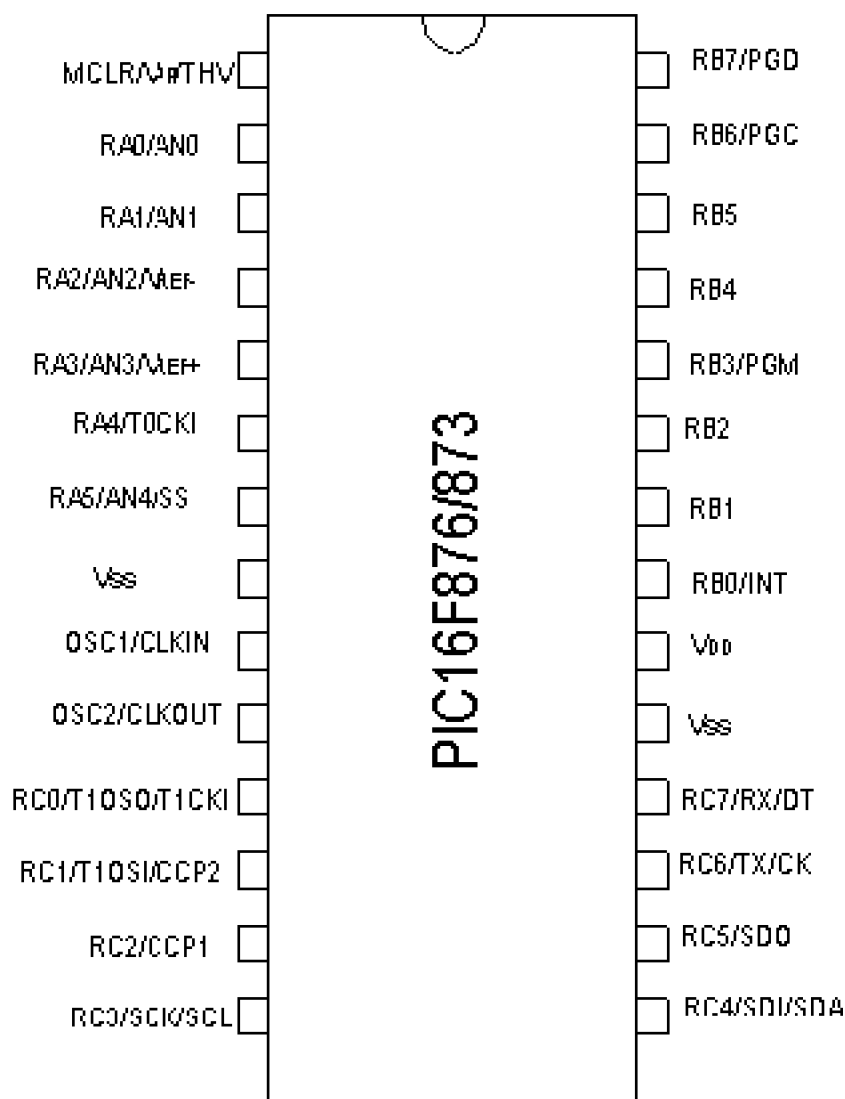
Set de 35 instrucciones en lenguaje ensamblador (bajo nivel), estas instrucciones son compatibles con todas las otras gamas de microcontroladores producidos por Microchip.

Otras funciones especiales: Timers, modulo de PWM, Perro guardián (Watch dog timer), posibilidades de comunicación en forma serial y paralela.

Tabla 4.1 Características Principales de Cuatro Microcontroladores pic

Características Principales PICmicro Rango Medio	PIC 16F873	PIC16F874	PIC16F876	PIC16F877
Frecuencia de Operación	DC-20MHz	DC-20MHz	DC-20MHz	DC-20MHz
Resets (y Retardos)	POR,BOR (PWRT,OST)	POR,BOR (PWRT,OST)	POR,BOR (PWRT,OST)	POR,BOR (PWRT,OST)
Memoria de Programa tipo FLASH (palabras de 14 bits)	4K	4K	8K	8K
Memoria de Datos (bytes)	192	192	368	368
Memoria de Datos EEPROM	128	128	256	256
Interrupciones	13	14	13	14
Puertos I/O (entrada/salida)	Puertos A,B,C	Puertos A,B,C,D,E	Puertos A,B,C	Puertos A,B,C,D,E
Timers	3	3	3	3
Modulos Capture/Compare/PWM	2	2	2	2
Comunicaciones Seriales	MSSP,USART	MSSP,USART	MSSP,USART	MSSP,USART
Comunicaciones Paralelas	--	PSP	--	PSP
Modulo de 10-bit Analogo-a-Digital	5 canales de entrada	8 canales de entrada	5 canales de entrada	8 canales de entrada
Conjunto de Instrucciones	35 Instrucciones	35 Instrucciones	35 Instrucciones	35 Instrucciones

La tabla 4.1 muestra un cuadro comparativo entre las principales características de funcionamiento de cuatro referencias de microcontroladores PIC pertenecientes a la gama media-alta.



En la figura 4.1 se presenta la distribución de pines del circuito integrado

Distribución de Pines del PIC16F876

El CI tiene 28 pines distribuidos de la siguiente forma:

Puerto A: Este puerto es incompleto, es decir, no tiene los 8 pines para completar un byte, solo tiene 6 pines (RA0, RA1, RA2, RA3, RA4 y RA5) que pueden ser configurados para trabajar como entrada o salida y en forma digital o análoga (tiene cinco canales análogos : AN0, AN1, AN2, AN3 y AN4) todos los pines de este puerto son multiplexados, es decir tienen funciones diversas según la configuración que se haga en el inicio del programa.

Puerto B: Este puerto es de 8 pines (RB0 a RB7) y algunos de los pines tienen funciones adicionales que se utilizan durante la programación o el depuramiento en circuito (In Circuit Debugging) del CI.

Puerto C: También está completo (RC0 a RC7) y todos sus pines son multiplexados, según la configuración hecha pueden operar para las funciones especiales de PWM,

comunicaciones y temporizadores especiales.

Pines de Alimentación: Son los pines VDD(+5V) y VSS(GND) , utiliza 3 pines para esta función:

Pines del Oscilador: Aunque pueden utilizarse varias configuraciones de oscilador, usualmente se utiliza un cristal que va entre los pines OSC1 y OSC2.

Pin del Reset: Este pin solo tiene esa función durante la ejecución de un programa, sin embargo, durante la programación en este pin del CI se aplica uno de los dos voltajes requeridos.

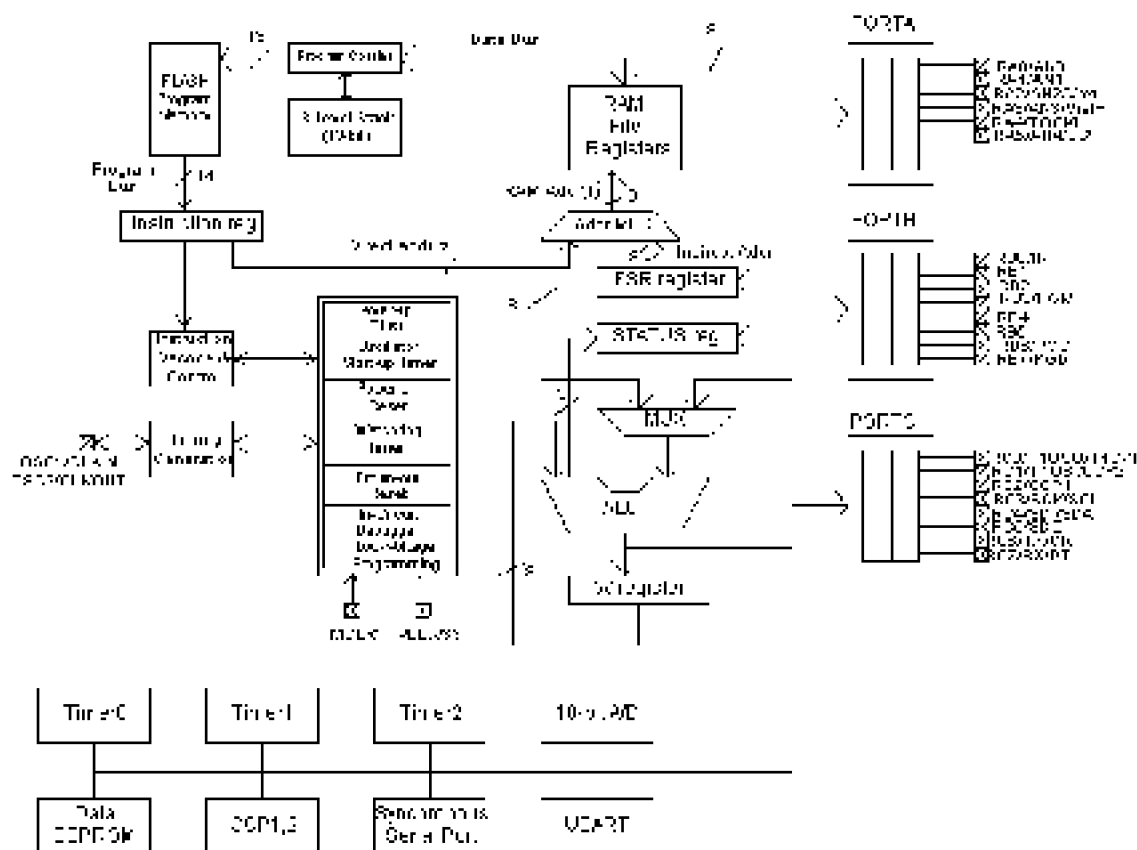
La tabla 4.2 muestra el resumen de las funciones de los pines del Microcontrolador PIC 16F876

Nombre del Pin	Tipo I/O/P (I=Entrada, O=Salida, P=Alimentación)	Descripción
OSC1/CLKIN	I	Entrada para Cristal Oscilador/Entrada de Reloj externo
OSC2/CLKOUT	O	Salida para Cristal Oscilador
MCLR/VPP/THV	I/P	Reset/Voltaje de Programación/Alto Voltaje en Control de Modo de Prueba
RA0/AN0 RA1/AN1 RA2/AN2/VREF- RA3/AN3/VREF+ RA4/AN4/PC4 RA5/SS/AN4	I/O	I/O.(Entradas-Salidas) RA0 puede ser también entrada analógica 0 RA1 puede ser también entrada analógica 1 RA2 puede ser también entrada analógica 2 o Voltaje negativo de referencia analógica. RA3 puede ser también entrada analógica 3 o Voltaje positivo de referencia analógica. RA4 también puede ser la entrada de reloj para el modulo <i>Timer0</i> . RA5 también puede ser la entrada analógica 4 o el <i>select</i> de <i>slave</i> del puerto serial síncrono
RB0/INT RB1 RB2 RB3/PGM RB4 RB5 RB6/PC0 RB7/PC1	I/O	I/O.(Entradas-Salidas) RB0 puede ser también el pin de interrupción externa. RB3 puede ser también la entrada para programación con bajo voltaje. RA4 también puede ser pin de interrupción por cambio de estado. RB5 también puede ser pin de interrupción por cambio de estado. RB6 también puede ser pin de interrupción por cambio de estado. Pin para señal de reloj durante programación serial. RB6 también puede ser pin de interrupción por cambio de estado. Pin para señal de datos durante programación serial
RC0/T1OSO/T1CKI RC1/T1OSI/CCP2 RC2/CCP1 RC3/SS/PC5 RC4/SDI/SDA RC5/SDO RC6	I/O	I/O.(Entradas-Salidas) RC0 puede ser también el pin de salida del Oscilador del Timer1 o la entrada de Reloj del Timer1. RC1 puede ser también el pin de entrada del Oscilador del Timer1 o Captura 2 entrada/Compara2 salida/Salida2 de PWM. RC2 puede ser también la Salida 1 de PWM/Captura1 entrada/Compara1 salida RC3 puede ser también la entrada/salida de reloj serial para los modos SPI e I ² C. RC4 también puede ser la entrada de datos en el modo SPI o salida/entrada de datos en el modo I ² C. RC5

Nombre del Pin	Tipo I/O/P (I=Entrada, O=Salida, P=Alimentación)	Descripción
		también puede ser la salida de datos en modo SPI RC6 también puede ser el pin de transmisión asíncrona USART o reloj síncrono. RC7 también puede ser el pin de recepción asíncrona USART o pin de datos síncronos.
Vss		Referencia de tierra para estados lógicos y pines I/O
VDD		Positivo para estados lógicos y pines I/O

Tabla 4.2 Resumen de Funciones de los Pines del Pic 16F876

En la figura 4.2 se muestra el esquema general de la arquitectura interna de este tipo de microcontrolador.



Arquitectura Interna del Microcontrolador pic16f876

4.2 PROGRAMACION DE MICROCONTROLADORES DE 8 BITS

Como sucede con otras marcas, para los microcontroladores pic también existen diversas posibilidades de programación:

Bajo nivel: Utilizando el lenguaje de 35 instrucciones llamado *ensamblador*. En este tipo de programación la escritura de código es bastante extensa y requiere de especial cuidado con la sintaxis y un profundo conocimiento de la arquitectura del microcontrolador. La principal ventaja es la eficiencia en el uso de la capacidad disponible de memoria de instrucciones.

Alto Nivel: Existen muchos tipos de Compiladores para pic en lenguaje C y en Basic principalmente. La ventaja es la reducción en la escritura de código y la versatilidad de estos tipos de lenguajes.

La tabla 4.3 presenta el resumen de las 35 instrucciones del lenguaje ensamblador y su sintaxis:

Conjunto de Instrucciones en Lenguaje Ensamblador para PIC

Operandos Mnemónicos	Descripción	Ciclos	Status Afectados
OPERACIONES SOBRE REGISTROS QUE SON ORIENTADAS A BYTES			
ADDWF f,d ANDWF f,d CLRF f CLRW COMF f,d DECF f,d DECFSZ f,d INCF f,d INCFSZ f,d IORWF f,d MOVF f,d MOVWF f NOP RLF f,d RRF f,d SUBWF f,d SWAPF f,d XORWF f,d	Suma W y f Operación AND de W con f Poner en 0 el registro f Poner en 0 el registro W Complementa f Decrementa f Decrementa f salta si f es cero Incrementa f Incrementa f salta si f es cero Or inclusiva entre W y f Mueve el registro f al destino d Mueve el contenido de W a f No operar Rota a la izquierda Rota a la derecha Sustraer W de f Intercambia los nibbles en f Or exclusiva entre W y f	1 1 1 1 1 1(2)	C1, DC, Z Z Z Z Z Z C C C,
OPERACIONES SOBRE REGISTROS QUE SON ORIENTADAS A BITS			
BCF f,b BSF f,b BTFSC f,b BTFSS f,b	Pone en 0 el bit b de f Pone en 1 el bit b de f Chequea el bit b de f. Salta si es cero Chequea el bit b de f. Salta si es uno	1 1 1(2) 1(2)	
OPERACIONES LITERALES Y DE CONTROL			
ADDLW k ANDLW k CALL k CLRWD GOTO k IORLW k MOVLW k RETFIE RETLW k RETURN SLEEP SUBLW k XORLW k	Suma el literal k con W Operación AND de k con W Llama subrutina Reset del Perro Guardián Ir a la dirección Or Inclusiva entre k y W Mueve k a w Regreso de una interrupción Regresar con literal en w Regreso de una subrutina Ir a modo de espera Sustraer W del literal k Or exclusiva entre W y k	1 1 2 1 2	C1, DC, Z Z TO, PD Z TO, PD C, DC, Z Z

El lenguaje utilizado para desarrollar esta aplicación es un tipo de BASIC especialmente adecuado para PIC (*PICbasic*). La relación en cuanto a la cantidad de código escrito en uno u otro lenguaje es de 1:5 o 1:4, es decir se requiere de 4 o cinco veces mas de código en lenguaje ensamblador para realizar la misma tarea que se obtiene programando en *Picbasic*. A continuación se presenta un resumen de las instrucciones utilizadas en el programa.

4.3 REFERENCIA DE DECLARACIONES UTILIZADAS:

CLEAR Hace cero todas las variables

END Detiene la ejecución e ingresa en modo de baja potencia

HIGH Hace alto la salida del pin

IF..THEN..ELSE..ENDIF Ejecuta declaraciones en forma condicional

LCDOUT Muestra caracteres en un LCD

LOW Hace bajo la salida de un pin

PAUSE Pausa (resolución 1mseg.)

PAUSEUS Pausa (resolución 1 □seg.)

PWM Salida modulada en ancho de pulso a un pin de salida

Algunos elementos, como el oscilador y las ubicaciones de los pin LCD, están predefinidas en *PicBasic*.

DEFINE le permite a un programa cambiar sus definiciones si así lo desea.

OSCdefine el tipo de Oscilador usado

3 para oscilador de 3.58 Mhz

4para oscilador de 4 Mhz

8para oscilador de 8 Mhz

10para oscilador de 10Mhz

12 para oscilador de 12 Mhz

16para oscilador de 16 Mhz

20para oscilador de 20 Mhz

Define puede ser usado para cambiar el valor predefinido del oscilador y las ubicaciones de los pines del LCD además de otras cosas. Estas definiciones deben estar en mayúsculas.

DEFINE LCD_DREG PORTB ´ puerto de datos del LCD

DEFINE LCD_DBIT 0 ´ datos LCD comenzando en bit 0 o 4

DEFINE LCD_RSREG PORTB ´ puerto de selección de registro LCD

DEFINE LCD_RSBIT 4 ´ bit de selección de registro del LCD

DEFINE LCD_EREG PORTB ´ puerto de habilitación del LCD

DEFINE LCD_EBIT 5 ´ bit de habilitación del LCD

DEFINE LCD_BITS 4 ´ tamaño en bits del bus de datos del LCD

DEFINE LCD_LINES 2 ´ numero de líneas del LCD

DEFINE OSC 4 ' 3 4 8 10 12 16 20 definición del oscilador.

Los comentarios se acompañan de ' o de ;

La sintaxis para los valores numéricos que se especifican en el programa es la siguiente:

% para valores binarios y \$ para valores en hexadecimal, por ejemplo:

%100 ' valor binario para el decimal 4.

\$100 ' valor hexadecimal para el decimal 256.

4.3.1 OPERADORES MATEMÁTICOS

Todas las operaciones matemáticas se realizan en orden jerárquico. Esto significa que existe precedencia para los operadores. Multiplicación y división son efectuados antes que la suma y la resta. Para asegurarse que las operaciones son efectuadas en el orden que se desea se usa paréntesis para agrupar las operaciones. Por ejemplo esta es una operación matemática con variables.

$$A = (B + C) * (D - E)$$

Todas las operaciones matemáticas se realizan sin signo y con una precisión de 16 bits.

Los operadores que soporta el lenguaje son:

+Suma

- Resta

* Multiplicación

** 16 bits superiores de la multiplicación

*/ 16 bits medios de la multiplicación

/ División

// Resto (módulo)

<< Desplazamiento izquierdo

>> Desplazamiento derecho

ABS Valor absoluto

COS Coseno

DCD 2m decodificador

DIG Dígito

MAX Máximo *

MIN Mínimo *

NCD Codificar

REv Invertir bits

SIN Seno

SQR Raíz cuadrada

& Bit inteligente AND

÷ Bit inteligente OR

^ Bit inteligente EXCLUSIVE OR

~ Bit inteligente NOT

& / Bit inteligente NOT AND

÷ / Bit inteligente NOT OR

4.3.1.1 MULTIPLICACION

Se efectúan multiplicaciones de 16 x 16 bits .El operador '*' devuelve los 16 bit inferiores del resultado de 32 bit. Esta es la multiplicación típica encontrada en los lenguajes de programación. El operador '**' devuelve los 16 bit superiores del resultado de 32 bit. Estos dos operadores pueden ser utilizados en conjunto para realizar multiplicaciones de 16 x 16 bit que produzcan resultados de 32 bit .

W1 = W0 * 1000 ´ multiplica el valor de W0 por 1000 y coloca el resultado en W1

W2 = W0 ** 1000 ´ W0 por 1000 y coloca los 16 bit superiores (que deben ser 0) en W2

El operador '*' ´ los 16 bit medios del resultado de 32 bit.

W3 = W1 * / W0 ´ multiplica W1 por W0 y coloca los 16 bit medios en W3

4.3.1.2 DIVISION

Se efectúan divisiones de 16 x 16 bit . El operador '/' devuelve el resultado de 16 bit . El operador '//' devuelve el resto (módulo del número).

W1 = W0 / 1000 ´ Divide el valor de W0 por 1000 y coloca el resultado en W1

W2 = W0 // 1000 ´ Divide el valor de W0 por 1000 y coloca el resto en W2

4.3.1.3 MAX y MIN

MAX y MIN devuelven el máximo y mínimo, respectivamente, de dos números. Se usan normalmente para limitar números a un valor, pero también se pueden utilizar con dos variables.

B1 = B0 MAX 100 ´ le asigna a B1 el mayor entre B0 y 100 (B0 debe estar entre 100 y 255)

B1 = B0 MIN 100 ´ le asigna a B1 el menor entre B0 y 100 (B1no puede ser mayor de 100)

4.3.2 OPERADORES DE COMPARACION

Se usan en declaraciones IF ... THEN para comparar una expresión con otra .Los operadores soportados

son :

= o == Igual

<> o != Diferente

< Menor

> Mayor

<= Menor o igual

>= Mayor o igual

Ejemplo:

If i > 10 then loop

4.3.3 OPERADORES LOGICOS

Los operadores lógicos entregan un resultado CIERTO /FALSO de su operación. El valor 0 es tratado como falso. Cualquier otro valor es cierto. Se usan junto a operadores de comparación en una declaración IF .. THEN .Los operadores soportados son :

AND o && AND lógico

OR o || OR lógico

XOR o ^ ^ OR exclusivo lógico

NOT AND NAND lógico

NOT OR NOR lógico

NOT XOR NXOR lógico

Ejemplo:

If (A == big) AND (B > mean) then run

4.4 OTRAS DECLARACIONES UTILIZADAS

LCDOUT

Muestra mensajes en un visor de cristal líquido inteligente (LCD). Se pueden manejar módulos LCD con un controlador Hitachi 44780 o equivalente. Estos LCD, usualmente, tienen 14 o 16 pines simples o duales en un extremo.

Un programa debe esperar, por lo menos, medio segundo antes de enviar el primer comando a un LCD. Puede tomar bastante tiempo a un LCD arrancar. Los comandos son enviados al LCD, enviando un \$FE seguido por el comando. Algunos comandos útiles son los siguientes:

\$FE, 1 Limpia visor

\$FE, 2 Vuelve a inicio (comienzo de la primera línea)

\$FE, \$0C Cursor apagado

\$FE, \$0E Subrayado del cursor activo

\$FE, \$0F Parpadeo del cursor activo

\$FE, \$10 Mueve cursor una posición hacia la izquierda

\$FE, \$14 Mueve cursor una posición hacia la derecha

\$FE, \$C0 Mueve cursor al comienzo de la segunda línea

Por ejemplo:

LCDOUT \$FE, \$C0

Hace que el visor comience a escribir caracteres en el principio de la segunda línea.

LCDOUT \$FE, 1, "Hello" 'limpia el visor y muestra "Hello"

El LCD puede estar conectado al microcontrolador Pic, usando un bus de 4 bits o uno de 8 bits. Si se usa un bus de 8 bits, todos los 8 bits deben estar en un puerto. Si se usa un bus de 4 bits, debe estar conectado a los 4 bits inferiores o a los 4 bits superiores de un puerto. Enable y Register Select deben estar conectados a algún pin de uno de los puertos. R/W debe estar colocado a tierra, ya que el comando de LCDOUT solamente es de grabación.

PAUSE

PAUSE Periodo

Detiene el programa por Periodo el cual se expresa en milisegundos. Periodo tiene 16 bits, por lo que los retardos pueden ser de hasta 65.535 milisegundos (un poco mas de 1 minuto). Esta instrucción no coloca el microcontrolador en modo de baja potencia como las otras funciones de retardo (NAP y SLEEP). Inclusive, consume mayor potencia, pero es más exacto. PAUSE asume una frecuencia de 4 Mhz en el oscilador. Si se usa un oscilador de otra frecuencia, se debe indicar usando el comando DEFINE OSC.

Ejemplo:

PAUSE 1000 'pausa de 1 segundo

4.5 CONVERSION ANALOGA/DIGITAL

Para poder procesar los datos correspondientes a la temperatura del elemento que se desea controlar se requiere un conversor análogo/digital. El microcontrolador pic utilizado posee esta función lo cual facilita las cosas y no incluye retardos adicionales en el proceso ya que el conversor esta integrado al microcontrolador.

4.5.1 CONFIGURACION DE REGISTROS DEL MODULO A/D

Se utilizó el canal 0 de conversión A/D con una resolución de 8 bits. Para poder utilizar satisfactoriamente esta función del microcontrolador se tuvieron que configurar de manera adecuado los registros adcon0 y adcon1 que son los que administran los aspectos relacionados con A/D.

El registro Adcon0 maneja los aspectos fundamentales como el reloj y la habilitación del proceso de conversión.

Estructura Interna del Registro ADCON0

R/W R/W R/W R/W R/W R/W U R/W							
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
bit7 bit0							R = Bit leíble W = bit escribible U= No implementado
bits 7 y 6		ADCS1:ADCS0: A/D Bits de Selección del Reloj de Conversión 00=FOSC/2 01=FOSC/8 10=FOSC/32 11=FRC (Reloj Derivado de un Oscilador RC)					
bits 5 al 3		CHS2:CHS0: bits de selección del canal análogo 000=canal 0 (RA0/AN0) 001=canal 1 (RA1/AN1) 010=canal2 (RA2/AN2) 011=canal3(RA3/AN3) 100=canal4(RA5/AN4)					
bit 2		GO/DONE: bit del status de la conversion A/D Si ADON=1 1=conversion A/D en progreso 0=conversion A/D detenida					
bit 1		No implementado, se lee como cero					
bit 0		ADON:bit de A/D activo 1=El modulo conversor A/D esta en operación 0=El modulo conversor A/D está inactivo y no consume corriente de operación.					

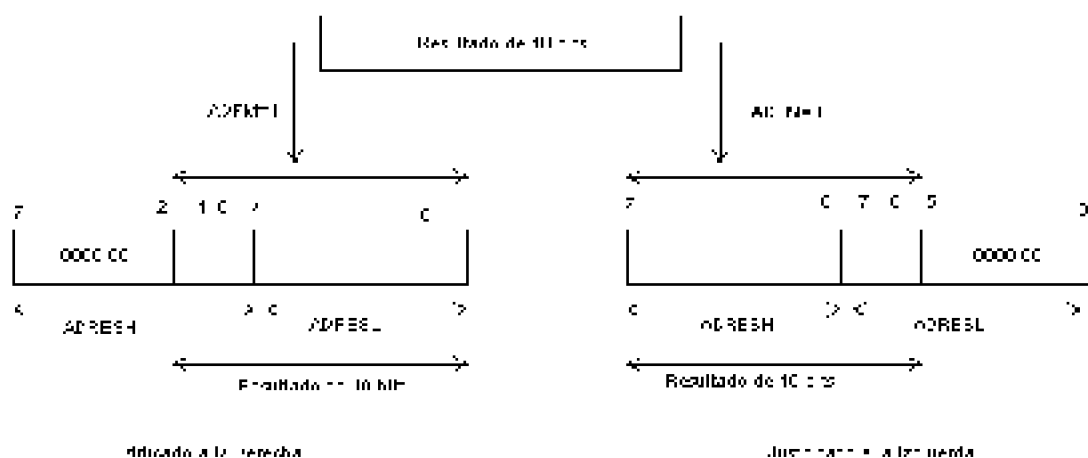
En el registro adcon1 se seleccionan los canales A/D que se van a utilizar y el formato de manejo de los datos.

Tabla 4.5 Estructura Interna del Registro ADCON1

CONTROL DIFUSO DE TEMPERATURA PARA CARGA RESISTIVA CON MICROCONTROLADOR DE 8 BITS

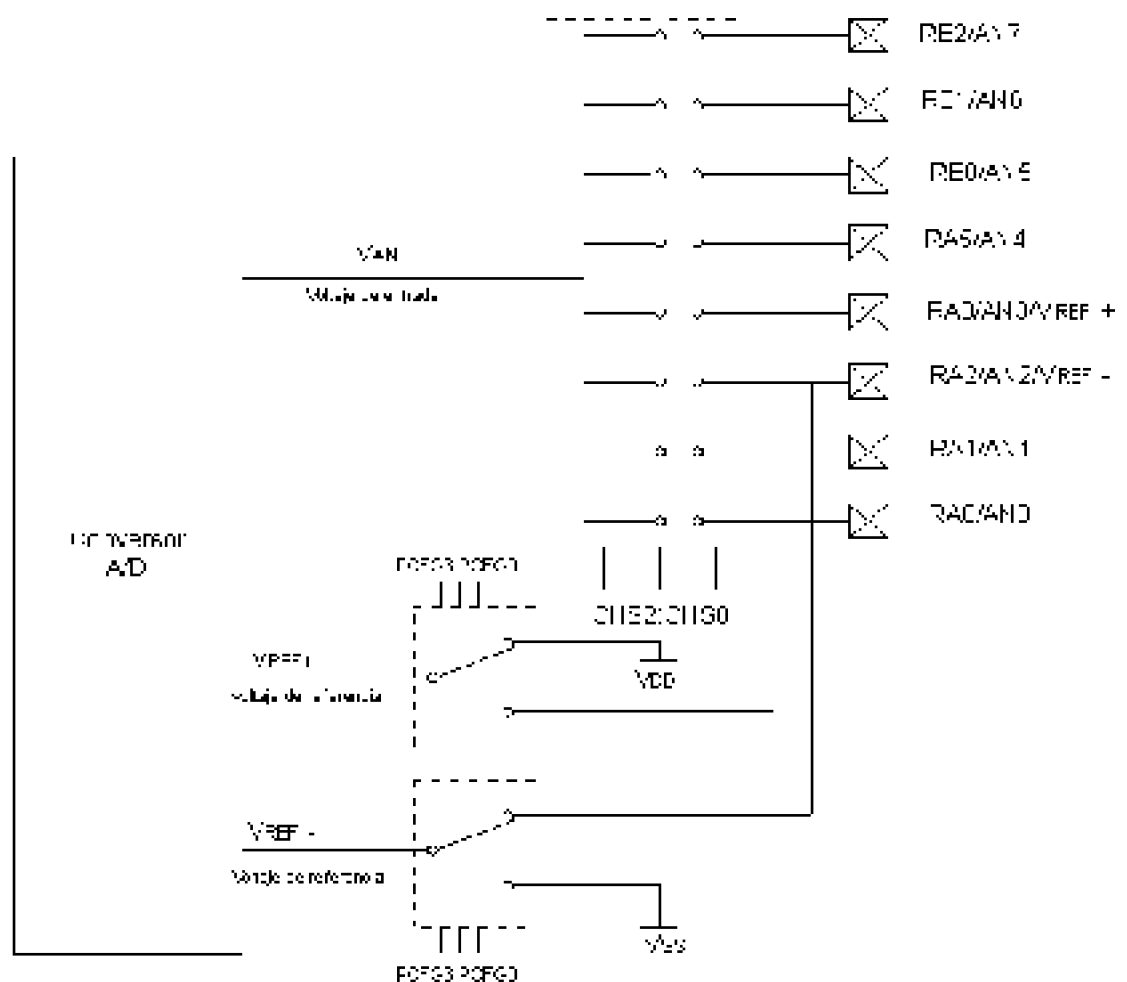
R/W U U U R/W R/W R/W R/W														
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0	R = Bit leíble W = bit escribible U= No implementado						
bit7bit0														
bit 7: ADFM: Selección del formato del resultado de A/D 1 = Justificado a la derecha. Los 6 bits mas significativos de ADRESH se leen como 0. 0 = Justificado a la izquierda. Los 6 bits menos significativos de ADRESH se leen como 0. bits 6 al 4: No implementados bits 3 al 0: PCFG3 : PCFG0 son los bits de control de configuración del puerto A/D														
PCFG3:	PCFG2:	PCFG1:	PCFG0:	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	CHAN/Refs
0000	A	A	A	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	A	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	A	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	A	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	A	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	A	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	A	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	A	A	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	A	A	A	RA3	RA2	1/2
A = entrada análoga D = I/O (entrada/salida) digital Nota: 1. Los canales AN5, AN6 y AN7 no están disponibles en los modelos de 28 pines. 2.La ultima columna hace la relación entre canales disponibles como entradas A/D y el numero de canales análogos usados como entradas de voltajes de referencia.														

La figura 4.3 muestra el manejo de los datos de la conversión según sea el formato seleccionado.



Manejo de los Resultados Según la Configuración del Registro

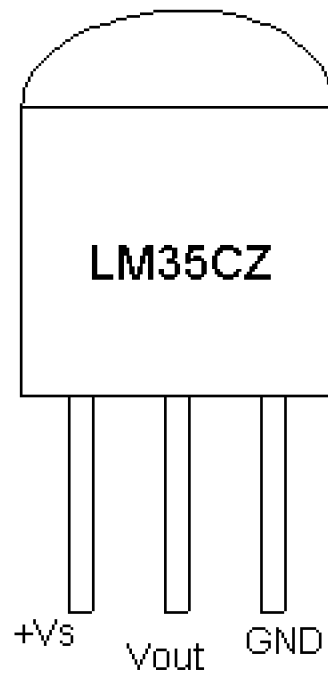
La figura 4.4 muestra el esquema general del modulo A/D del pic



Estructura General del Modulo A/D del PIC16F876/77

4.6 SENSOR DE TEMPERATURA

El sensor utilizado es el LM35DZ de National Semiconductor el cual entrega 10mV por cada grado centígrado y tiene un rango de operación de -55 a 150 grados Celsius.



Distribución de Pines del Sensor LM35

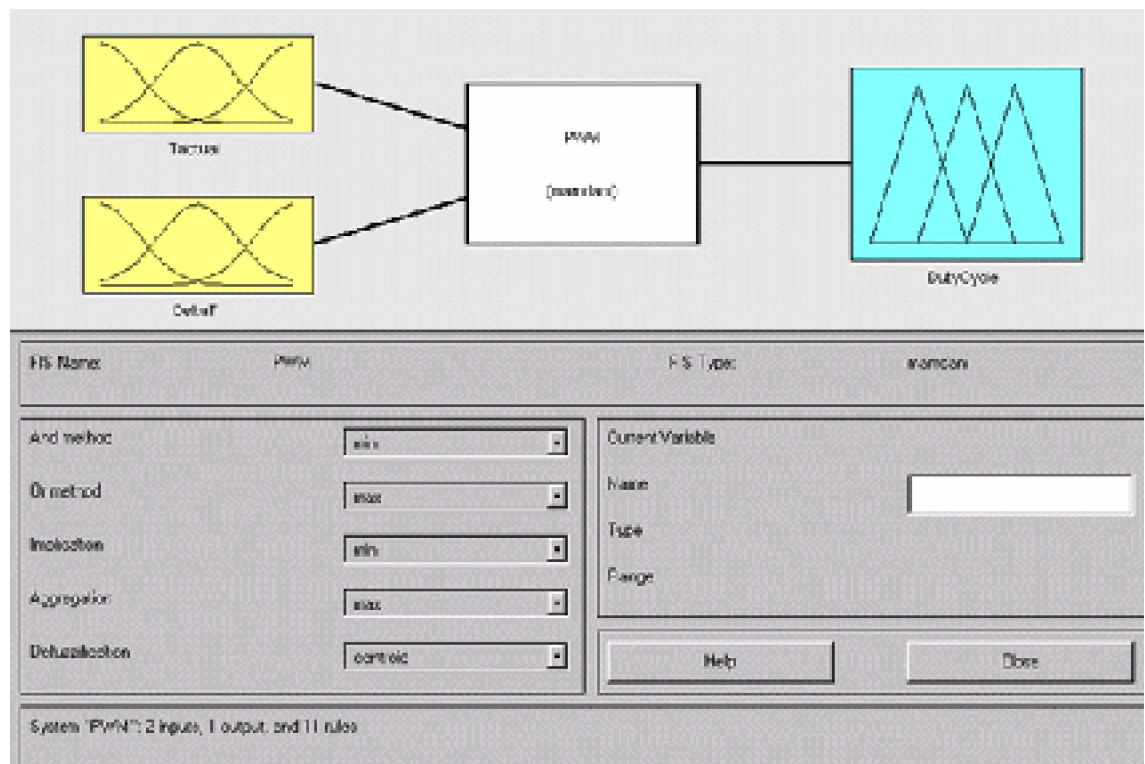
Características Principales del Sensor:

- Calibrado directamente en ° Celsius (Centígrados)
- Lineal, + 10.0 mV/°C factor de escala
- 0.5°C de precisión garantizada (a +25°C)
- Caracterizado para el rango completo de -55° a +150°
- Apropiado para aplicaciones de medición remota
- Trabaja con voltajes desde 4 hasta 30 voltios
- Consume menos de 60 μ A
- Bajo autocalentamiento, 0.08°C en aire estático
- Presenta una No linealidad de solo $\pm 1/4^\circ\text{C}$ (valor típico)
- Salida de baja impedancia, 0.1 W para una carga de 1 mA load

5. MEMORIAS DE DISEÑO

5.1 MODELADO DIFUSO DEL SISTEMA

En la figura 5.1 se muestra la presentación general del sistema difuso que se simuló en MATLAB. Este tiene 2 variables de entrada y una variable de salida.



Sistema Difuso

5.1.1 FUNCIONES DE PERTENENCIA Y UNIVERSOS DE DISCURSO

La temperatura se modeló mediante tres funciones de pertenencia en un universo de discurso de 0 a 100 grados centígrados.

Para las funciones de pertenencia de la variable de entrada Tactual se definieron las siguientes rectas:

Recta1:

$$Y1=m1T+b \quad m1=1/5 \quad 0=-6+b \quad b=6 \quad Y1=(-1/5)T+6 \quad Y1=-0.2T+6$$

Para efectos de manipulación matemática dentro del controlador, ya que este no maneja números con cifras decimales ni números negativos.

Y1=-200T + 6000 Recta2: $Y2=m2T +b \quad m2=1/15 \quad 0=1.67+b \quad b=-1.67$
 $Y2=(1/15)T-1.67 \quad Y2=0.067T-1.67$

Para efectos de manipulación matemática dentro del controlador:

Y2=67T-1670

Recta3:

$$Y3=m3T+b \quad m3=-1/20 \quad 0=-3+b \quad \text{de donde } b=3 \quad Y3=(-1/20)T+3 = -0.05T+3$$

Para efectos de manipulación matemática dentro del controlador:

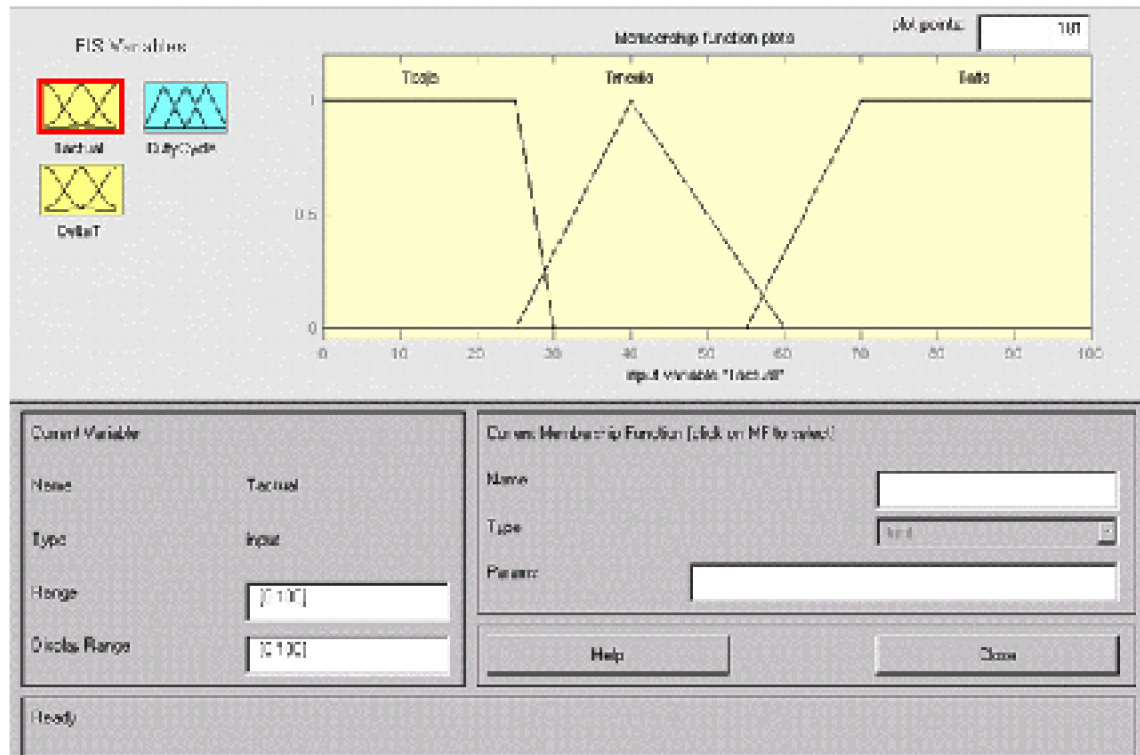
Y3=-50 T + 3000 Recta4:

$$Y4=m4T +b \quad m4=1/15 \quad 0=-11/3+b \quad \text{de donde } b=11/3 \quad Y4=(1/15)T-11/3 = 0.067D -$$

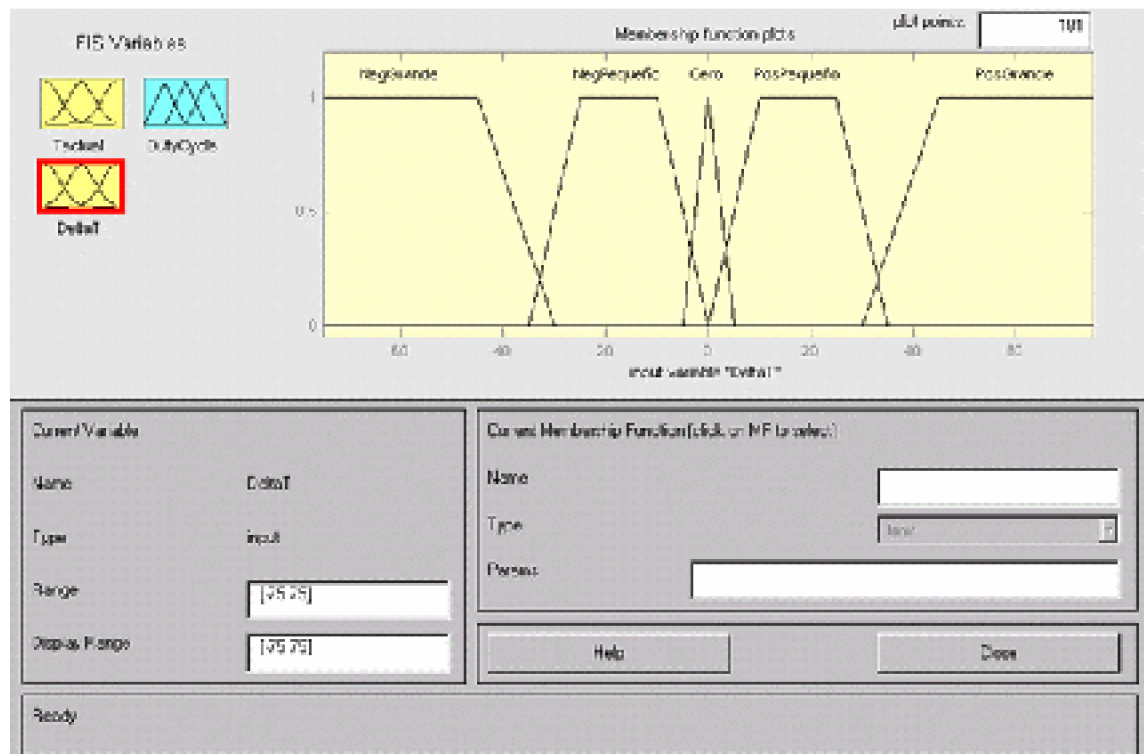
3.67

Para efectos de manipulación matemática dentro del controlador:

Y4=67T-3670



Funciones de Pertenencia y Universo de Discurso para la Temperatura



Funciones de Pertenencia y Universo de Discurso para Delta de Temperatura

Para las funciones de pertenencia de la variable de entrada DeltaT se definieron las siguientes rectas:

Recta1:

$$Y1 = m1DT + b \quad m1 = -1/15 \quad 0 = -40/15 + b \quad b = 2.67 \quad Y1 = (-1/15)DT + 2.67$$

$$Y1 = -0.067DT + 2.67$$

Para efectos de manipulación matemática dentro del controlador:

$$Y1 = -67DT + 2670$$

Recta2:

$$Y2 = m2DT + b \quad m2 = 1/10 \quad 0 = 3.5 + b \quad b = -3.5 \quad Y2 = (1/10)DT - 3.5 \quad Y2 = 0.1DT - 3.5$$

Para efectos de manipulación matemática dentro del controlador:

$$Y2 = 100DT - 3500$$

Recta3:

$$Y3=m3DT+b \quad m3=-1/10 \quad 0 = -7 + b \text{ de donde } b=7 \quad Y3=(-1/10)DT + 7 = -0.1DT + 7$$

Para efectos de manipulación matemática dentro del controlador:

$$Y3=-100 DT + 7000$$

Recta4:

$$Y4=m4DT + b \quad m4= 1/5 \quad 0 =13 + b \text{ de donde } b=-13 \quad Y4=(1/5)DT -13 = 0.2DT - 13$$

Para efectos de manipulación matemática dentro del controlador:

$$Y4=200DT-13000$$

Recta5: $Y5=m5DT+b \quad m5=-1/5 \quad 0 =-15+ b \text{ de donde } b=15 \quad Y5=(-1/5)DT+15 =-0.2DT+15$

Para efectos de manipulación matemática dentro del controlador:

$$Y5=-200DT+15000$$

Recta6:

$$Y6=m6DT+b \quad m6=1/10 \quad 0=7+b \text{ de donde } b=-7 \quad Y6=(1/10)DT -7 =0.1DT -7$$

Para efectos de manipulación matemática dentro del controlador:

$$Y6=100DT -7000$$

Recta7:

$Y7=m7DT + b \quad m7=-1/10 \quad 0 =-10.5+ b \text{ de donde } b=10.5 \quad Y7=(-1/10)DT+10.5 =-0.1DT+10.5$

Para efectos de manipulación matemática dentro del controlador:

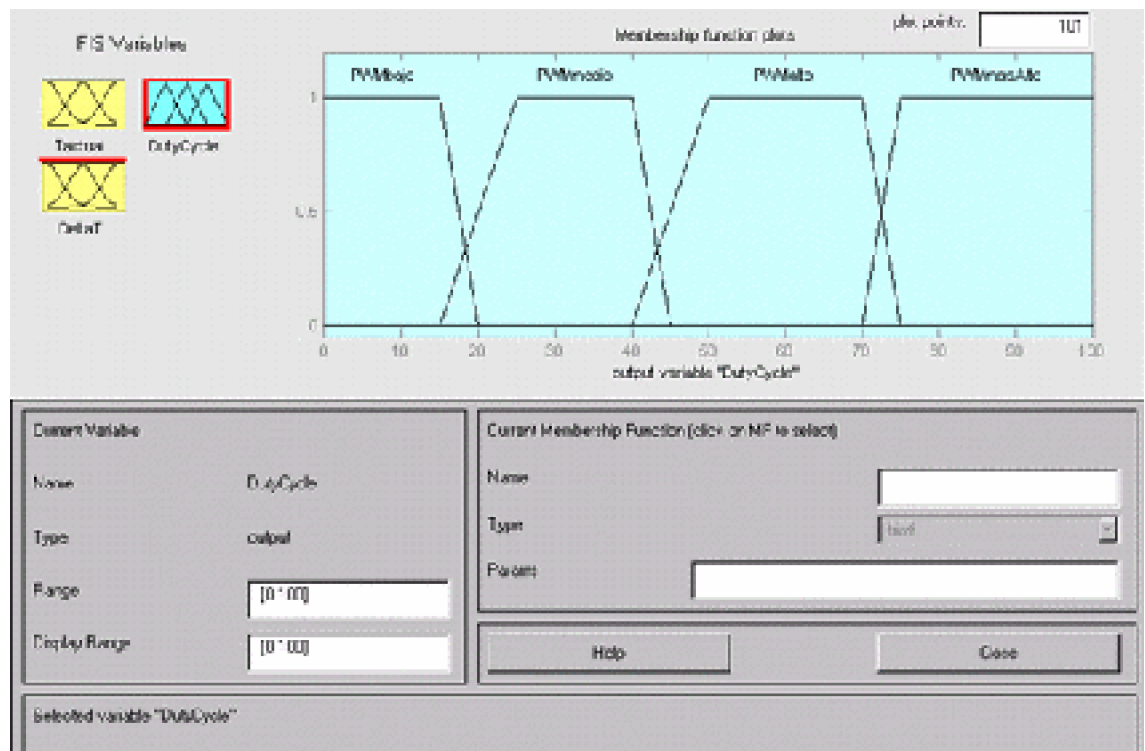
$$Y7=-100DT +10500$$

Recta8:

$Y8=m8DT + b \quad m8=1/15 \quad 0 =6.67+ b \text{ de donde } b=6.67 \quad Y8=(1/15)DT -6.67 =0.067DT -6.67$

Para efectos de manipulación matemática dentro del controlador:

$$Y8=67DT -6670$$



Funciones de Pertenencia y Universo de Discurso para DutyCycle

Para las funciones de pertenencia de la variable de salida se definieron las siguientes rectas:

Recta1:

$$Y1=m1PW+b$$

$$m1=-1/5$$

$$0 = -20/5 + b \quad b=4$$

$$Y1=(-1/5)PW+4$$

$$Y1=-0.2PW+4$$

Para efectos de manipulación matemática dentro del controlador:

$$Y1=-200PW+4000$$

Recta2:

$$Y2=m2PW+b$$

$$m2=1/10$$

$$0 = 1.5 + b \quad b=-1.5$$

$$Y2=(1/10)PW-1.5$$

$$Y2=0.1PW-1.5$$

Para efectos de manipulación matemática dentro del controlador:

$$Y2=100PW-1500$$

Recta3:

$$Y3=m3PW+b$$

$$m3=-1/5$$

$$0 = -9 + b \text{ de donde } b=9$$

$$Y3=(-1/5)PW+9 = -0.2PW+9$$

Para efectos de manipulación matemática dentro del controlador:

$$\mathbf{Y3=-200PW+9000}$$

Recta4:

$$Y4=m4PW+b$$

$$m4=1/10$$

$$0 = 4 + b \text{ de donde } b=-4$$

$$Y4=(1/10)PW-4 = 0.1PW-4$$

Para efectos de manipulación matemática dentro del controlador:

$$\mathbf{Y4=100PW-4000}$$

Recta5:

$$Y5=m5PW+b$$

$$m5=-1/5$$

$$0 = -15 + b \text{ de donde } b=15$$

$$Y5=(-1/5)PW+15 = -0.2P+15$$

Para efectos de manipulación matemática dentro del controlador:

$$\mathbf{Y5=-200PW+15000}$$

Recta6:

$$Y6=m6PW+b$$

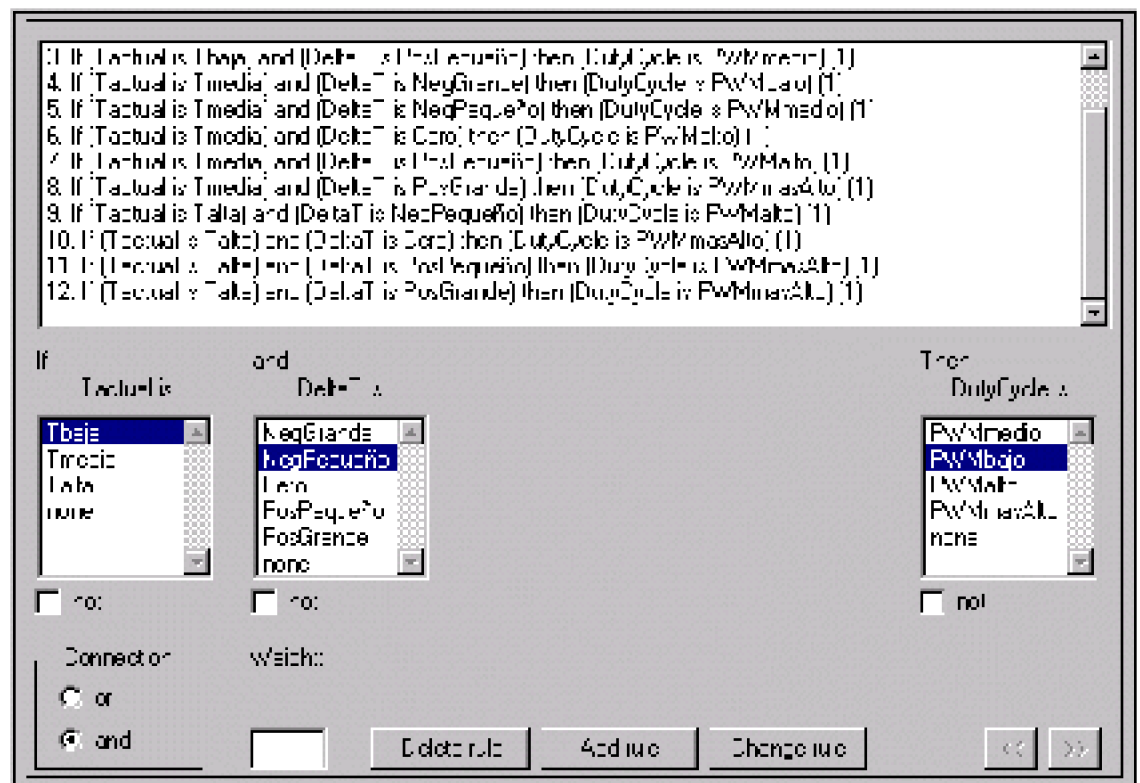
$$m6=1/5$$

$$0 = 14 + b \text{ de donde } b=-14$$

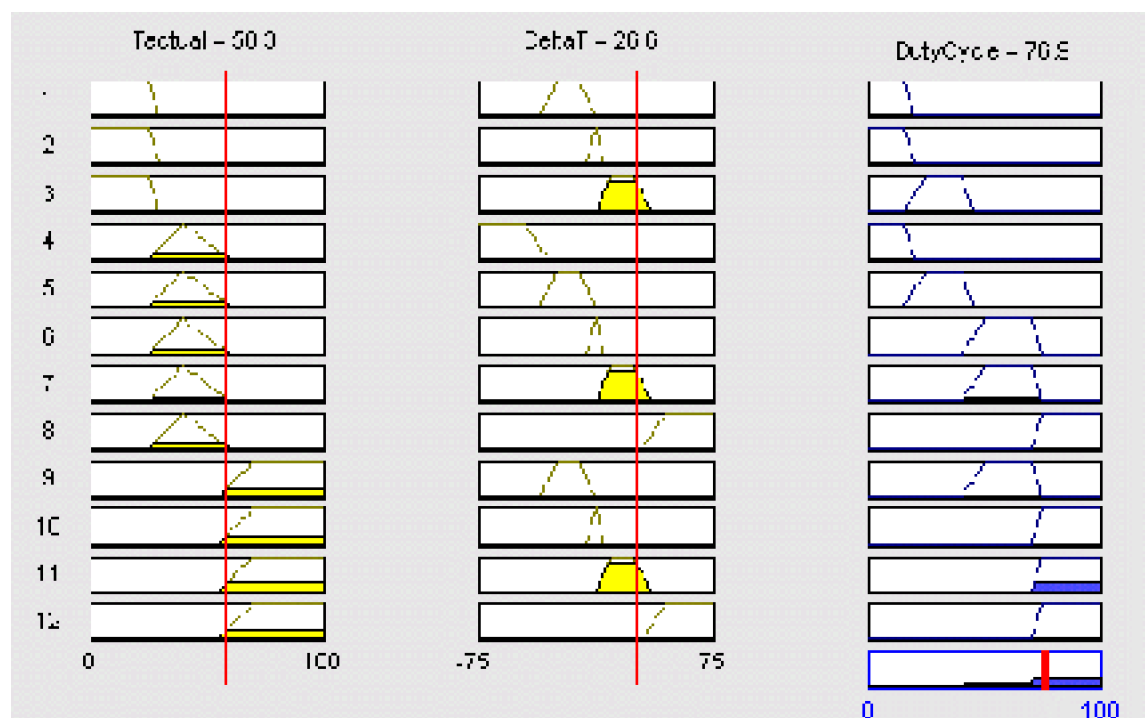
$$Y6=(1/5)PW+14 = 0.2PW+14$$

Para efectos de manipulación matemática dentro del controlador:

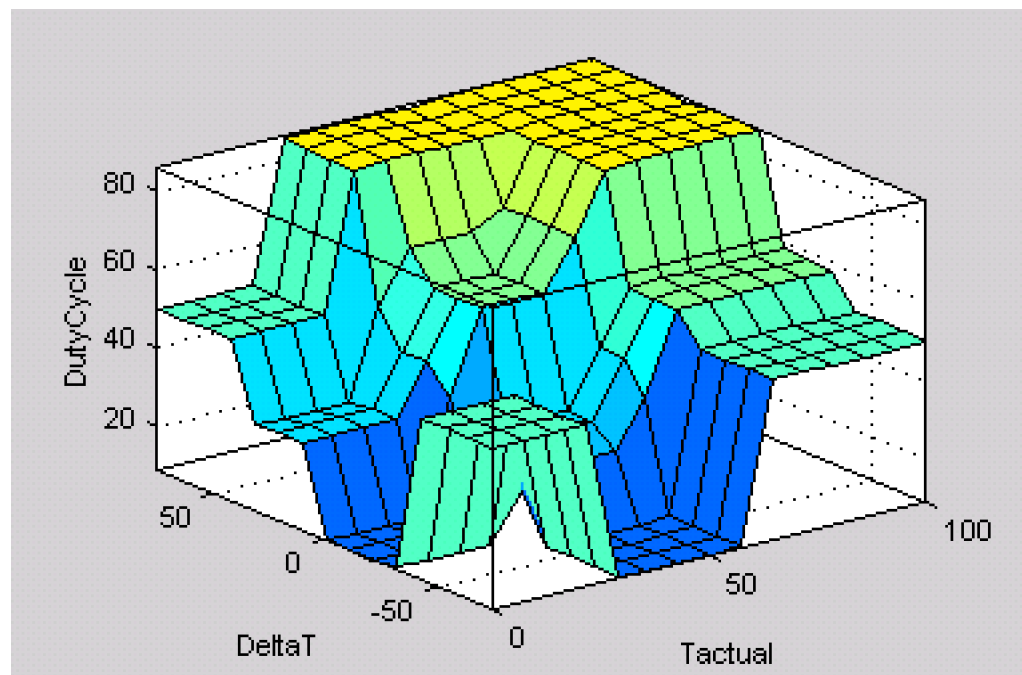
$$\mathbf{Y6=200PW-14000}$$



Vista del Editor de Reglas de MATLAB-FIS



Visor de Reglas



Superficie de Representación de la Base de Reglas

5.2 ESTRATEGIA DE CONTROL

Se trata de la elección del universo de discurso, la elección de las variables y el conjunto de reglas, el método de inferencia, la forma de la acción de control.

Preliminarmente se han visualizado las siguientes condiciones sobre las cuales se simuló en MATLAB:

Variables de entrada:

- Temperatura actual
- Delta de Temperatura (ΔT)

Variable de Salida (acción de control)

- Porcentaje de Ciclo Útil en PWM sobre la velocidad del Ventilador

Método de Inferencia Difusa:

- Mamdani
- Método AND = mínimo
- Método OR = máximo
- Implicación $\square$ mínimo
- Agregación $\square$ máximo

- Método de defuzificación □ □ centroide

Universo de discurso y Funciones de Pertenencia

Temperatura actual:

- Esta variable estará entre un rango de 0 a 100 grados centígrados
- El universo de discurso de esta variable se cubrirá con tres términos: Temperatura baja, Temperatura media y Temperatura alta representadas con dos funciones de pertenencia trapezoidales y una función triangular.

Delta de Temperatura (□T):

- Se resumirán los posibles cambios en el valor de la temperatura con cinco términos: Negativo Grande Negativo Pequeño Cero Positivo Pequeño Positivo Grande, representados por cuatro funciones de pertenencia trapezoidales y una función triangular.
- El rango de □T es de un -75% a un 75% aproximadamente y será calculado de la siguiente forma: $\square T = \frac{(T_{actual} - T_{anterior}) \times 100\%}{T_{actual}}$

Porcentaje de Ciclo Útil en PWM sobre la velocidad del Ventilador:

- El rango será 0 a 100%
- Por razones de longitud del nombre, la acción de control se cubrirá con cuatro términos: PWM bajo PWM medio PWM alto PWM mas alto, representados por cuatro funciones de pertenencia trapezoidales.

Base de Reglas Difusa:

Esta base estará constituida por doce reglas de operación del tipo : **If**Tactual es A **and** DeltaT es B **Then** Duty Cycle is C

5.3 IMPLEMENTACION DEL CONTROLADOR DIFUSO

Se desarrollara el sistema de control difuso compuesto por el banco de resistencias, el microcontrolador de 8 bits, el sensor integrado de temperatura, la pantalla de cristal liquido (LCD) como elemento de visualización y un ventilador con motor cc de 12 V. El software de control traduce su acción mediante la variación con PWM (Modulación por Ancho de Pulso) de la velocidad del ventilador.

5.3.1 CONJUNTO DE REGLAS DEL CONTROLADOR

R1: si Tactual es Tbaja y DeltaT es NegPequeño **entonces** DutyCycle es PWMbajo

R2: si Tactual es Tbaja y DeltaT es Cero **entonces** DutyCycle es PWMbajo

R3: si Tactual es Tbaja y DeltaT es PosPequeño **entonces** DutyCycle es PWMmedio

R4: si Tactual es Tmedia y DeltaT es NegGrande **entonces** DutyCycle es PWMbajo

R5: si Tactual es Tmedia y DeltaT es NegPequeño **entonces** DutyCycle es PWMmedio

R6: si Tactual es Tmediay DeltaT es Cero **entonces** DutyCycle es PWMalto

R7: si Tactual es Tmedia y DeltaT es PosPequeño **entonces** DutyCycle es PWMalto

R8: si Tactual es Tmedia y DeltaT es PosGrande **entonces** DutyCycle es PWMmasalto

R9: si Tactual es Talta y DeltaT es NegPequeño **entonces** DutyCycle es PWMalto

R10: si Tactual es Talta y DeltaT es Cero **entonces** DutyCycle es PWMmasalto

R11: si Tactual es Talta y DeltaT es PosPequeño **entonces** DutyCycle es PWMmasalto

R12: si Tactual es Talta y DeltaT es PosGrande **entonces** DutyCycle es PWMmaslto

Estas doce reglas también se pueden presentar en forma de matriz:

DT T	Neg Grande	Neg Pequeño	Cero	Pos Pequeño	Pos Grande
baja		BAJO	BAJO	MEDIO	
media	BAJO	MEDIO	ALTO	ALTO	MAS ALTO
alta		ALTO	MAS ALTO	MAS ALTO	MAS ALTO

Matriz de Entradas con Acción de Control

5.3.2 INFERENCIA DE LA ACCION DE CONTROL

Para una temperatura A y un delta de Temperatura C y una acción de control B

Temperatura =T, Delta de Temperatura=DT

$$\square B(\text{PWM}) = (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{bajaxNegpeq} \square \text{bajo} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{bajaxcero} \square \text{bajo} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{bajaxpospeq} \square \text{medio} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{mediaxNeggrande} \square \text{bajo} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{mediaxNegpeq} \square \text{medio} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{mediaxcero} \square \text{alto} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{mediaxpospeq} \square \text{alto} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{mediaxposgrande} \square \text{masalto} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{altaxNegpeq} \square \text{alto} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{altaxcero} \square \text{masalto} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{altaxpospeq} \square \text{masalto} (T, DT, \text{PWM}) \})$$

$$U (\sup T, DT \{ [\square A(T) * \square C(DT)] * \square \text{altaxposgrande} \square \text{masalto} (T, DT, \text{PWM}) \})$$

Al desarrollar el operador “sup” de la expresión anterior se obtiene:

$$\square B(\text{PWM}) = (\sup T \{ \square A(T) * \square \text{baja}(T) \} * \sup DT \{ \square C(DT) * \square \text{negpeq}(DT) \} * \square \text{bajo} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{baja}(T) \} * \sup DT \{ \square C(DT) * \square \text{cero}(DT) \} * \square \text{bajo} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{baja}(T) \} * \sup DT \{ \square C(DT) * \square \text{pospeq}(DT) \} * \square \text{medio} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{media}(T) \} * \sup DT \{ \square C(DT) * \square \text{neggrande}(DT) \} * \square \text{bajo} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{media}(T) \} * \sup DT \{ \square C(DT) * \square \text{negpeq}(DT) \} * \square \text{medio} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{media}(T) \} * \sup DT \{ \square C(DT) * \square \text{cero}(DT) \} * \square \text{alto} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{media}(T) \} * \sup DT \{ \square C(DT) * \square \text{pospeq}(DT) \} * \square \text{alto} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{media}(T) \} * \sup DT \{ \square C(DT) * \square \text{posgrande}(DT) \} * \square \text{masalto} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{alta}(T) \} * \sup DT \{ \square C(DT) * \square \text{negpeq}(DT) \} * \square \text{alto} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{alta}(T) \} * \sup DT \{ \square C(DT) * \square \text{cero}(DT) \} * \square \text{masalto} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{alta}(T) \} * \sup DT \{ \square C(DT) * \square \text{pospeq}(DT) \} * \square \text{masalto} (\text{PWM}))$$

$$U (\sup T \{ \square A(T) * \square \text{alta}(T) \} * \sup DT \{ \square C(DT) * \square \text{posgrande}(DT) \} * \square \text{masalto} (\text{PWM}))$$

Si se continúa con el desarrollo de las expresiones de acuerdo a los métodos elegidos para los operadores:

$$\square B(\text{PWM}) = \max T (\min [\square A(T), \square \text{baja}(T)]) * \max DT (\min [\square C(DT), \square \text{negpeq}(DT)]) * \square \text{bajo} (\text{PWM})$$

$$U \max T (\min [\square A(T), \square \text{baja}(T)]) * \max DT (\min [\square C(DT), \square \text{cero}(DT)]) * \square \text{bajo} (\text{PWM})$$

$$U \max T (\min [\square A(T), \square \text{baja}(T)]) * \max DT (\min [\square C(DT), \square \text{pospeq}(DT)]) * \square \text{medio} (\text{PWM})$$

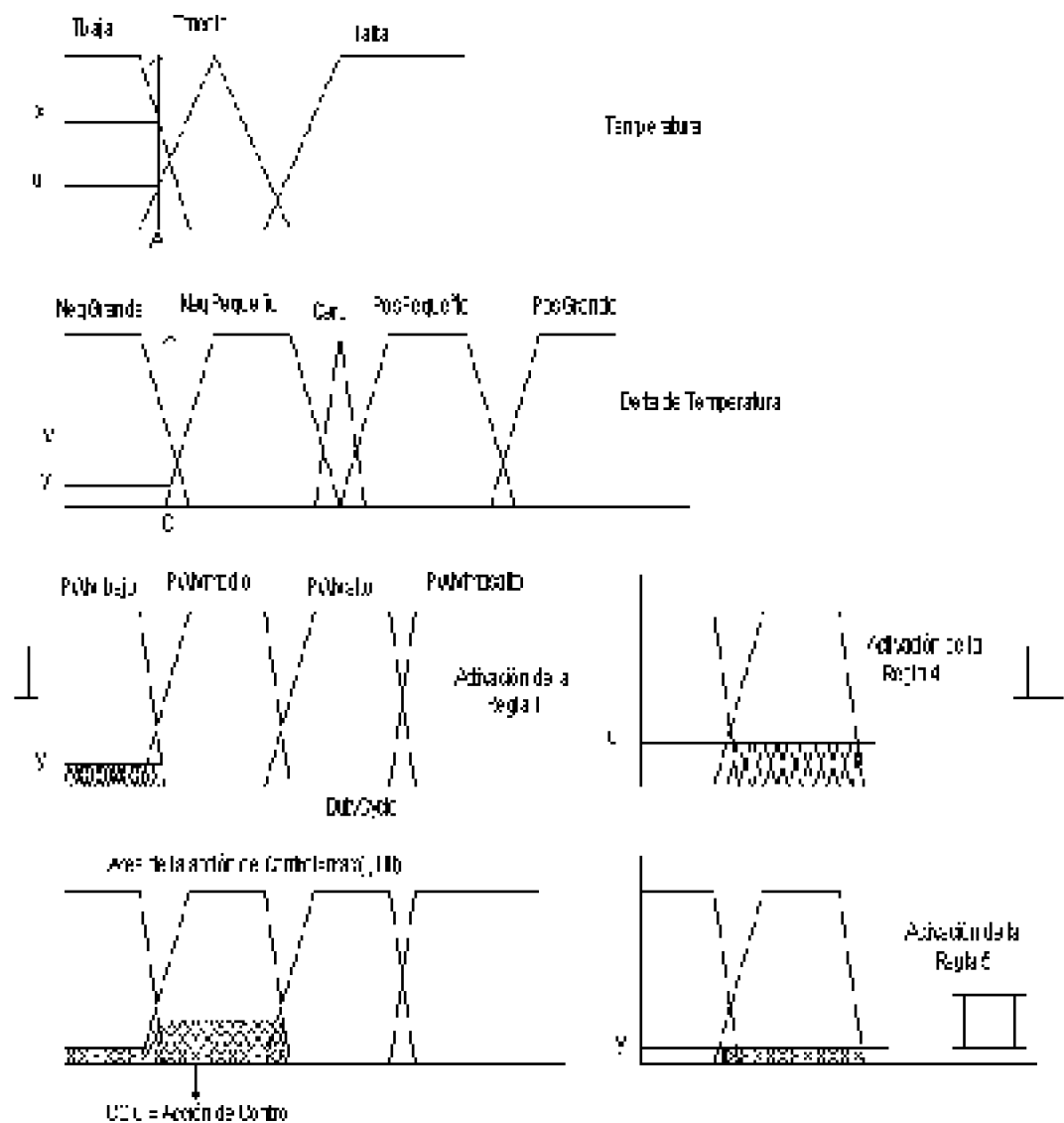
U

$$\max T (\min [\square A(T), \square \text{media}(T)]) * \max DT (\min [\square C(DT), \square \text{neggrande}(DT)]) * \square \text{bajo} (\text{PWM})$$

$$U \max T (\min [\square A(T), \square \text{media}(T)]) * \max DT (\min [\square C(DT), \square \text{negpeq}(DT)]) * \square \text{medio} (\text{PWM})$$

$$\begin{aligned}
& U \max T(\min[\square A(T), \square media(T)]) * \max DT(\min[\square C(DT), \square cero(DT)]) * \square alto(PWM) \\
& U \max T(\min[\square A(T), \square media(T)]) * \max DT(\min[\square C(DT), \square pospeq(DT)]) * \square alto(PWM) \\
& U \max T(\min[\square A(T), \square media(T)]) * \max DT(\min[\square C(DT), \square posgrande(DT)]) * \square masalto(PWM) \\
& U \max T(\min[\square A(T), \square alta(T)]) * \max DT(\min[\square C(DT), \square negpeq(DT)]) * \square alto(PWM) \\
& U \max T(\min[\square A(T), \square alta(T)]) * \max DT(\min[\square C(DT), \square cero(DT)]) * \square masalto(PWM) \\
& U \max T(\min[\square A(T), \square alta(T)]) * \max DT(\min[\square C(DT), \square pospeq(DT)]) * \square masalto(PWM) \\
& U \max T(\min[\square A(T), \square alta(T)]) * \max DT(\min[\square C(DT), \square posgrande(DT)]) * \square masalto(PWM) \\
& \text{Al terminar de desarrollar estas expresiones se tiene:} \\
& \square B(PWM) = \\
& \min\{\max T(\min[\square A(T), \square baja(T)]), \max DT(\min[\square C(DT), \square negpeq(DT)]) \square bajo(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square baja(T)]), \max DT(\min[\square C(DT), \square cero(DT)]) \square bajo(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square baja(T)]), \max DT(\min[\square C(DT), \square pospeq(DT)]) \square medio(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square media(T)]), \max DT(\min[\square C(DT), \square neggrande(DT)]) \square bajo(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square media(T)]), \max DT(\min[\square C(DT), \square negpeq(DT)]) \square medio(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square media(T)]), \max DT(\min[\square C(DT), \square cero(DT)]) \square alto(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square media(T)]), \max DT(\min[\square C(DT), \square pospeq(DT)]) \square alto(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square media(T)]), \max DT(\min[\square C(DT), \square posgrande(DT)]) \square masalto(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square alta(T)]), \max DT(\min[\square C(DT), \square negpeq(DT)]) \square alto(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square alta(T)]), \max DT(\min[\square C(DT), \square cero(DT)]) \square masalto(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square alta(T)]), \max DT(\min[\square C(DT), \square pospeq(DT)]) \square masalto(PWM)\} \\
& U \min\{\max T(\min[\square A(T), \square alta(T)]), \max DT(\min[\square C(DT), \square posgrande(DT)]) \square masalto(PWM)\}
\end{aligned}$$

La U hace referencia a un operador OR y por lo tanto la expresión final del conjunto de pertenencia de la acción de control es el máximo de la unión entre los resultantes de todos los términos.



Representación Gráfica del Proceso de Inferencia

5.3.3 ALGORITMO DE CONTROL

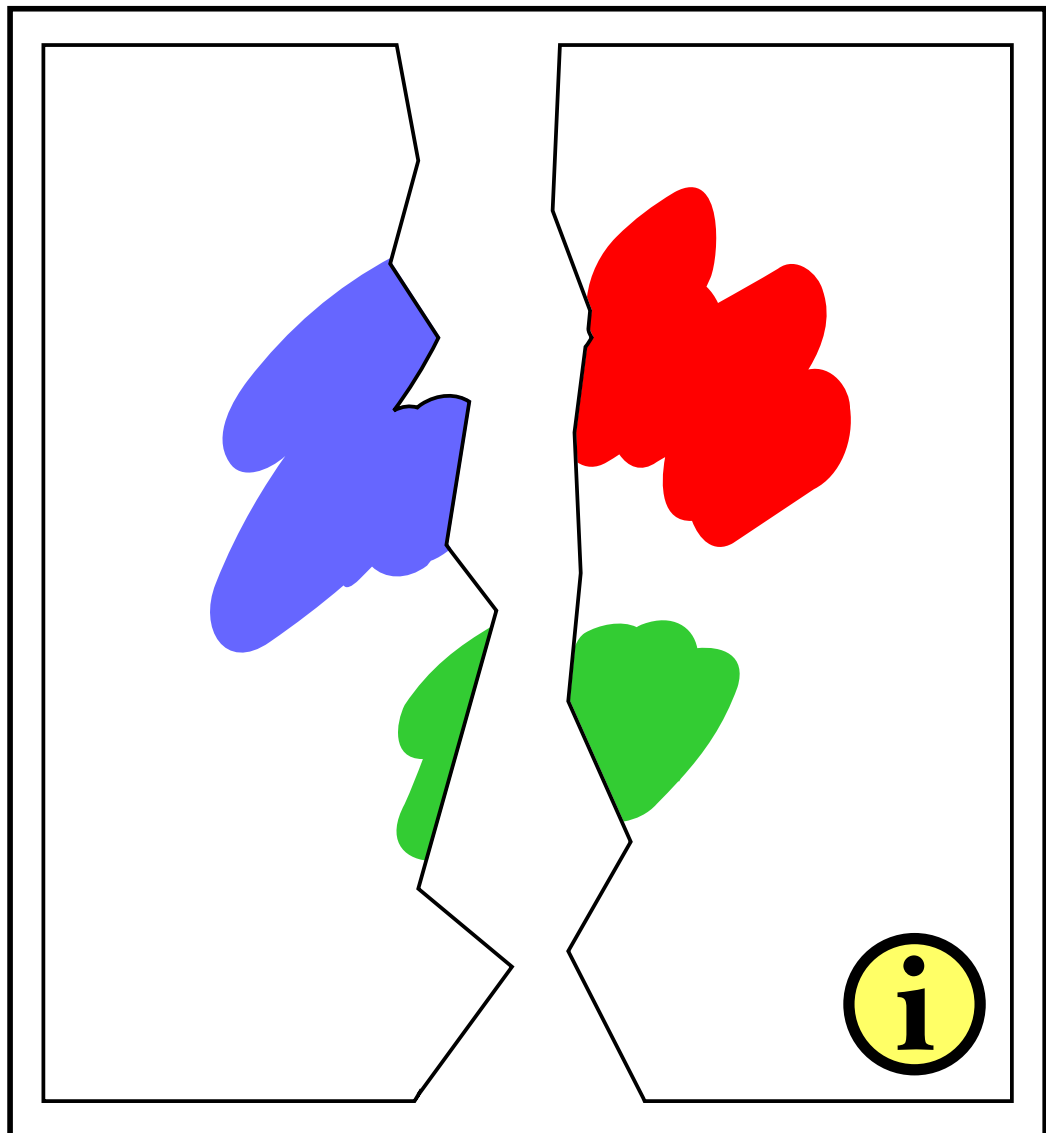


Diagrama de Flujo del Algoritmo de Control

Para el calculo de la acción de control se defuzificó la región conjunto de salida de Duty Cycle utilizando el método del Centro de Gravedad.

5.3.4 PROGRAMA PARA EL MICROCONTROLADOR

Este es el programa con el algoritmo de control implementado:

'* Name : fuzzycont.BAS *

'* Author : Gustavo A. Meneses B *

'* Date : 09/05/04 *

'* *

```
*****

Define osc 4 'definición de la frecuencia de trabajo (4MHz)
trisb=%00000000 'todo el puerto B se configura como salida
trisa=%00111111 'Los seis pines del puerto A se configuran como 'entradas
trisc=$00 'todo el puerto C se configura como salida
DefineADC_BITS8' conversión A/D con resolución de 8 bits
DefineADC_CLOCK 3' reloj interno para A/D conversión (3=rc)
DefineADC_SAMPLEUS50' tiempo de muestreo en uS
adcon1=%01111110 'Solo el pin 0 del puerto A es Análogo
ADCON0=%00000001 ' Configuración y habilitación del modulo A/D
DEFINE LCD_DREG PORTB 'Puerto de datos para el LCD
DEFINE LCD_DBIT 0 'Los datos empiezan desde el bit 0
DEFINE LCD_RSREG PORTC 'Puerto asignado al pin Register Select del LCD
DEFINE LCD_RSBIT 7 'bit asignado al pin Register Select (RS) del LCD
DEFINE LCD_EREG PORTC 'Puerto asignado al pin Enable del LCD
DEFINE LCD_EBIT 6 'bit asignado al pin Enable del LCD
DEFINE LCD_BITS 8 'bus de datos del LCD de 8 bits
DEFINE LCD_LINES 2 'Numero de líneas del LCD
DEFINE LCD_COMMANDUS 2000 'Retardo en microsegundos para los 'comandos
para el LCD
DEFINE LCD_DATAUS 50 'Retardo para los datos en microsegundos
T var byte'variable para almacenar el valor presente de la 'temperatura (tipo byte)
Ti var byte 'Variable: Temperatura preestablecida
DT var word 'Variable: Cambio de temperatura
Ubaja var word 'Variable: Grado de pertenencia a "Temperatura 'Baja"
Umedia var word 'Variable: Grado de pertenencia a "Temperatura Media"
Ualta var word 'Variable: Grado de pertenencia a "Temperatura Alta"
Ung var word 'Variable: Grado de pertenencia a "Cambio Negativo Grande"
Unp var word 'Variable: Grado de pertenencia a "Cambio Negativo Pequeño"
Uc var word 'Variable: Grado de pertenencia a "Cambio Cero"
Upp var word 'Variable: Grado de pertenencia a "Cambio Positivo Pequeño"
Upg var word 'Variable: Grado de pertenencia a "Cambio Positivo Grande"
a1 var word 'Variable: Mínimo entre Ubaja y Unp
a2 var word 'Variable: Mínimo entre Ubaja y Uc
```

a3 var word 'Variable: Mínimo entre Ubaja y Upp

a4 var word 'Variable: Mínimo entre Umedia y Ung

a5 var word 'Variable: Mínimo entre Umedia y Unp

a6 var word 'Variable: Mínimo entre Umedia y Uc

a7 var word 'Variable: Mínimo entre Umedia y Upp

a8 var word 'Variable: Mínimo entre Umedia y Upg

a9 var word 'Variable: Mínimo entre Ualta y Unp

a10 var word 'Variable: Mínimo entre Ualta y Uc

a11 var word 'Variable: Mínimo entre Ualta y Upp

a12 var word 'Variable: Mínimo entre Ualta y Upg

b1 var word 'Variable: Máximo entre a1 y a2

b2 var word 'Variable: Máximo entre a1, a2 y a4

b3 var word 'Variable: Máximo entre a3 y a5

b4 var word 'Variable: Máximo entre a6 y a7

b5 var word 'Variable: Máximo entre a6, a7 y a9

b6 var word 'Variable: Máximo entre a8 y a10

b7 var word 'Variable: Máximo entre a8, a10 y a11

b8 var word 'Variable: Máximo entre a8,a10, a11 y a12

Upwmbajo var word 'Variable: Grado de pertenencia a "PWM Bajo"

Upwmmedio var word 'Variable: Grado de pertenencia a "PWM Medio"

Upwmalto var word 'Variable: Grado de pertenencia a "PWM Alto"

Upwmmas var word 'Variable: Grado de pertenencia a "PWM más Alto"

PW var word 'Variable: Valor de la acción de control

Pa var word 'Valor de PW hasta donde la pertenencia vale bi para PWM 'Bajo y donde comienza la recta1

Pb var word 'Valor de PW donde la pertenencia comienza a valer bi para 'PWM Medio y donde termina la recta2

Pc var word 'Valor de PW hasta donde la pertenencia vale bi para PWM 'Medio y donde comienza la recta3

Pd var word 'Valor de PW donde la pertenencia comienza a valer bi para 'PWM Alto y donde termina la recta4

Pe var word 'Valor de PW hasta donde la pertenencia vale bi para PWM 'Alto y donde comienza la recta5

Pf var word 'Valor de PW donde la pertenencia comienza a valer bi para 'PWM mas Alto y donde termina la recta6

Upwm var word 'Máximo grado de pertenencia entre los términos 'Upwmbajo, Upwmmedio, Upwmalto y Upwmmas

Upwm1 var word 'Máximo grado de pertenencia entre los términos Upwmbajo y 'Upwmmedio

Upwm2 var word 'Máximo grado de pertenencia entre los términos Upwmalto y 'Upwmmas

A var word 'Grado de pertenencia de Upwm normalizado

B var word 'Acumulador de los grados de pertenencia al evaluar el COG

C var word 'producto Acción de control*Grado de Pertenencia al evaluar el COG

D var word 'Acumulador de los resultados de C

Act var word 'COG

accion var word 'para ajustar act al formato de PWM (act 100=255 PWM)

f var byte 'contador para pausa de 1 segundo

g var byte 'contador para pausa de 2 segundos

h var byte 'contador para pausa de 1/2 segundo

low portc.0 'inicio con estos pines en bajo (ninguna resistencia conectada)

low portc.1

low portc.2

low portc.3

low portc.4

clear 'todas las variables se inicializan en 0

Ti=15 ' Valor prefijado para los cálculos de inferencia difusa

explore: 'rutina para accionar las resistencias

if porta.1=0 then 'pin del pulsador correspondiente a R1

toggle portc.0

Lcdout \$fe, 1' Display en blanco

Lcdout "Resistencia 1" Mensaje del display

LCDOUT \$FE, \$C0

LCDOUT "carga 1"

gosub pausa1s 'ir a subrutina: pausa de 1 segundo

goto res1

else

goto res1

endif

```
pausa1s: 'retardo de 1 segundo
f=f+1
PWM portc.5,accion,1
if f=200 then
f=0
return
else
goto pausa1s
endif
res1:'rutina de conexión/desconexión de la resistencia 2
if porta.2=0 then
toggle portc.1
Lcdout $fe, 1' Display en blanco
Lcdout "Resistencia 2" Mensaje del display
LCDOUT $FE, $C0
LCDOUT "carga 2"
gosub pausa1s 'ir a hacer pausa de 1 segundo
goto res2
else
goto res2
endif
res2:'rutina de conexión/desconexión de la resistencia 3
if porta.3=0 then
toggle portc.2
Lcdout $fe, 1'Display en blanco
Lcdout "Resistencia 3" Mensaje del display
LCDOUT $FE, $C0
LCDOUT "carga 3"
gosub pausa1s
goto res3
else
goto res3
endif
res3:'rutina de conexión/desconexión de la resistencia 4
```

```
if porta.4=0 then
toggle portc.3
Lcdout $fe, 1' Display en blanco
Lcdout "Resistencia 4" Mensaje del display
LCDOUT $FE, $C0
LCDOUT "carga 4"
gosub pausa1s
goto res4
else
goto res4
endif
res4:rutina de conexión/desconexión de la resistencia 5
if porta.5=0 then
toggle portc.4
Lcdout $fe, 1' Display en blanco
Lcdout "Resistencia 5" Mensaje del display
LCDOUT $FE, $C0
LCDOUT "carga 5"
gosub pausa1s
goto Tiempo
else
goto Tiempo
endif
Tiempo: 'tiempo durante el cual se exploran las teclas (5 segundos)
e=e+1
WHILE e <= 100
PWM portc.5,accion,10
goto explore
WEND
e=0
goto CaptureT
CaptureT:rutina de adquisición de valor de temperatura
pause 500
ADCON0.2 = 1' Inicio de la conversión
```

```

notdone: Pause 5
If ADCON0.2 = 1 Then notdone' Espera un bajo en el bit-2 de ADCON0, 'conversion
terminada
T = ADRESH' Mueve el byte superior a la variable T
Goto CalcDT
pausa2s:
g=g+1
PWM portc.5,accion,2
if g=200 then
g=0
return
else
goto pausa2s
endif
CalcDT:
DT=70+T-Ti
Lcdout $fe, 1' display en blanco
Lcdout "DT: ", DEC DT' aparece el valor indicado en el display
LCDOUT $FE, $C0 'segunda linea del display
LCDOUT "T actual: ",DEC T 'aprece el valor indicado
gosub pausa2s 'ir a hacer pausa de 2 segundos
Goto ClasifDT
ClasifDT:'rutina de clasificación del valor de delta de T
if DT>=0 and DT<25 then
Ung=1000
Unp=0
Uc=0
Upp=0
Upg=0
goto ClasificarT
else
goto NegGra
endif
NegGra:' ¿ DT es negativo grande ?

```

```
if DT>=25 and DT<35 then
  Ung=-67*DT+2670
  Unp=0
  Uc=0
  Upp=0
  Upg=0
  goto ClasificarT
else
  goto NPeq
endif
NPeq: '¿DT es negativo pequeño ?
if DT>=35 and DT<=40 then
  Ung=-67*DT+2670 'rectas de la función de pertenencia
  Unp=100*DT-3500
  Uc=0
  Upp=0
  Upg=0
  goto ClasificarT
else
  goto NegPeq
endif
NegPeq:
if DT>40 and DT<=45 then
  Ung=0
  Unp=100*DT-3500 'rectas de la función de pertenencia DT negativo
  Uc=0 'pequeño
  Upp=0
  Upg=0
  goto ClasificarT
else
  goto NePeq
endif
NePeq:
if DT>45 and DT<=60 then 'evaluación de otro tramo del universo de
```

```
Ung=0' discurso DT negativo pequeño
Unp=1000
Uc=0
Upp=0
Upg=0
goto ClasificarT
else
goto NeP
endif
NeP:'evaluación del tramo final del universo de
if DT>60 and DT<=65 then'discurso DT negativo pequeño
Ung=0
Unp=-100*DT+7000
Uc=0
Upp=0
Upg=0
goto ClasificarT
else
goto LeftC
endif
LeftC:'parte izquierda de la función de pertenencia
if DT>65 and DT<=70 then' DT Cero
Ung=0
Unp=-100*DT+7000
Uc=200*DT-13000
Upp=0
Upg=0
goto ClasificarT
else
goto RightC
endif
RightC:'parte derecha de la función de pertenencia
if DT>70 and DT<=75 then'DT cero
Ung=0
```

```
Unp=0
Uc=-200*DT+15000
Upp=100*DT-7000
Upg=0
goto ClasificarT
else
goto PosPequeno
endif
PosPequeno: '¿DT es Positivo Pequeño?'
if DT>75 and DT<=80 then
Ung=0
Unp=0
Uc=0
Upp=100*DT-7000
Upg=0
goto ClasificarT
else
goto PosPeq
endif
PosPeq:
if DT>80 and DT<=95 then
Ung=0 'La función de pertenencia es trapezoidal
Unp=0
Uc=0
Upp=1000
Upg=0
goto ClasificarT
else
goto PosP
endif
PosP:
if DT>95 and DT<=100 then
Ung=0
Unp=0
```

```
Uc=0
Upp=-100*DT+10500'recta de la función de pertenencia
Upg=0' DT Positivo Pequeño
goto ClasificarT
else
goto PoP
endif
PoP:
if DT>100 and DT<=105 then
Ung=0
Unp=0
Uc=0
Upp=-100*DT+10500'evaluación del grado de membresía de DT
Upg=67*DT-6670
goto ClasificarT
else
goto PosGrande
endif
PosGrande:'¿ DT es Positivo Grande? '
if DT>105 and DT<=115 then
Ung=0
Unp=0
Uc=0
Upp=0
Upg=67*DT-6670
goto ClasificarT
else
Ung=0
Unp=0
Uc=0
Upp=0
Upg=1000'la función de membresía es trapezoidal
goto ClasificarT
endif
```

```
ClasificarT: 'rutina de clasificación de la temperatura
if T >= 0 and T < 25 then
    Ubaja = 1000 'asignación del grado de membresía
    Umedia = 0
    Ualta = 0
    goto MemberT
else
    goto Media
endif
Media: '¿T es media?
if T >= 25 and T < 30 then
    Ubaja = -200 * T + 6000
    Umedia = 67 * T - 1600
    Ualta = 0
    goto MemberT
else
    goto MMedia
endif
MMedia:
if T >= 30 and T <= 40 then
    Ubaja = 0
    Umedia = 67 * T - 1600 'recta de función de pertenencia TMedia
    Ualta = 0
    goto MemberT
else
    goto MMedia
endif
MMMedia:
if T > 40 and T <= 55 then
    Umedia = -50 * T + 3000 'asignación de grados de pertenencia
    Ubaja = 0
    Ualta = 0
    goto MemberT
else
```

```

goto Alta
endif
Alta: ' ¿ T es alta ?
if T>55 and T<=60 then
Umedia=-50*T+3000
Ualta=67*T-3670
Ubaja=0
goto MemberT
else
goto AAlta
endif
AAlta:
if T>60 and T<=70 then
Ubaja=0 'asignación de grados de membresía
Umedia=0
Ualta=67*T-3670
goto MemberT
else
Ubaja=0
Umedia=0
Ualta=1000 'la función de pertenencia es trapezoidal
goto MemberT
endif
MemberT:
goto InfMachine
InfMachine: 'Maquina de Inferencia difusa
a1 = Ubaja MIN Unp ' a1 es el mínimo de Ub y Unp
a2 = Ubaja MIN Uc
a3 = Ubaja MIN Upp
a4 = Umedia MIN Ung
a5 = Umedia MIN Unp
a6 = Umedia MIN Uc
a7 = Umedia MIN Upp
a8 = Umedia MIN Upg

```

```
a9 = Ualta MIN Unp
a10 = Ualta MIN Uc
a11 = Ualta MIN Upp
a12 = Ualta MIN Upg
ConjSalida:'Construcción del conjunto de inferencia difusa
b1 = a1 max a2 ' b1 es el mayor entre a1 y a2
b2 = b1 max a4 ' este es el máximo para PWM bajo
b3 = a3 max a5 ' este es el máximo para PWM medio
b4 = a6 max a7
b5 = b4 max a9 ' este es el máximo para PWM alto
b6 = a8 max a10
b7 = b6 max a11
b8 = b7 max a12' este es el máximo para PWM mas alto
CrearSalida:'Definición de rectas para conjunto de salida
Pa=(4000-b2)/200
Pb=(b3+1500)/100
Pc=(9000-b3)/200
Pd=(b5+4000)/100
Pe=(15000-b5)/200
Pf=(b8+14000)/200
iterar:'se hace un barrido sobre el universo de discurso
Lcdout $fe, 1'del conjunto de salida de %PWM
Lcdout "PW: ", DEC PW
LCDOUT $FE, $C0
LCDOUT "B: ", DEC B
gosub pausamed
PW=PW+2'resolución del barrido
if PW <= 100 then
goto Defuz
else
goto FINAL
endif
pausamed:'pausa de 250ms
h=h+1
```

```
PWM portc.5,accion,1
if h=50 then
h=0
return
else
goto pausamed
endif
Defuz:'clasificación de PW en las funciones de
if PW >=0 and PW<Pa then'pertenencia del conjunto de salida
Upwmbajo=b2
goto PWbajo
else
goto PWbajo
endif
PWbajo:' ¿ PW es bajo?
if PW >=Pa and PW<=20 then
Upwmbajo=-200*PW+4000
goto PWMed
else
Upwmbajo=0 ' no hay membresía en PWM bajo
goto PWMed'voy a buscar PWMmedio
endif
PWMed:'¿ PW es medio ?
if PW >=15 and PW<=Pb then
Upwmmedio=100*PW-1000 'recta perteneciente a la función de pertenencia
goto PWMedio
else
goto PWMedio
endif
PWMedio:
if PW>Pb and PW<=Pc then
Upwmmedio=b3 'la función de pertenencia es trapezoidal
goto PWMedright
else
```

```
goto PWMedright
endif
PWMedright: 'lado derecho de la función de pertenencia PWMmedio
if PW>Pc and PW<=45 then
Upwmmedio=-200*PW+9000
goto PWMal
else
Upwmmedio=0 ;nuevo
goto PWMal
endif
PWMal: 'lado izquierdo de la función de pertenencia PWMalto
if PW>=40 and PW<Pd then
Upwmalto=100*PW-4000
goto PWMalto
else
goto PWMalto
endif
PWMalto: 'parte central de la función de pertenencia PWMalto
if PW>=Pd and PW<Pe then
Upwmalto=b5
goto PWMalright
else
goto PWMalright
endif
PWMalright: 'parte derecha de la función de pertenencia PWMalto
if PW>=Pe and PW<=75 then
Upwmalto=-200*PW+15000
goto PWMmas
else
Upwmalto=0 ;nuevo
goto PWMmas
endif
PWMmas: 'parte izquierda de la función
if PW>=70 and PW<Pf then 'de pertenencia PWMmasalto
```

```

Upwmmas=200*PW-14000
goto PWMmasalto
else
goto PWMmasalto
endif
PWMmasalto:'parte final de la función
if PW>=Pf and PW<=100 then'de pertenencia PWMmas alto
Upwmmas=b8
goto Maxuniones
else
Upwmmas=0 'no hay pertenencia en PWMmasalto
goto Maxuniones
endif
Maxuniones:
Upwm1=Upwmbajo max Upwmmedio' Upwm es el máximo de
Upwm2=Upwmalto max Upwmmas'Upwmbajo, Upwmmedio
Upwm=Upwm1 max Upwm2'Upwmalto y Upwmmasalto
goto COG
COG:
A=Upwm/100'divido para no desbordar el acumulado A=10 es pertenencia 1
B=A+B'acumulador de la suma de los Upwm
C=PW*(Upwm/100)
D=C+D'acumulador de la suma del producto PW*Upwm
goto iterar
FINAL:'subrutina final con la acción de control
Act=(D/B)
Lcdout $fe, 1' display en blanco
Lcdout "D: ", DEC D' aparece el valor indicado
LCDOUT $FE, $C0
LCDOUT "B: ", DEC B
gosub pausa2s'va a hacer una pausa2s
Lcdout $fe, 1
Lcdout "Act: ", DEC Act' aparece el valor indicado
LCDOUT $FE, $C0

```

```
LCDOUT "valor de Act"
gosub pausa1s'va a hacer una pausa de 1s
if Act>75 then
accion=255'la acción máxima es 255 (va de 0 a 255)
goto control
else
goto conversion
endif
conversion:
accion=Act*3'factor de corrección para acción de control
goto control
control:
Lcdout $fe, 1' display en blanco
Lcdout "accion: ", DEC accion' aparece el valor indicado
LCDOUT $FE, $C0
LCDOUT "control""despliegue de la acción de control
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
PWM portc.5,accion,100
D=0
B=0
PW=0
Goto explore
End'FIN
```

5.3.5 PLANO ELECTRICO DEL MONTAJE

6. RESULTADOS

El objetivo del controlador es mantener la carga con temperaturas considerablemente más bajas que aquellas que resultarían bajo las mismas condiciones de operación pero en lazo abierto. Específicamente se busca establecer sobre la carga una temperatura que se aleje lo menos posible del promedio de la temperatura ambiente de la ciudad, la cual usualmente está en el rango de los 15°C a los 30°C dependiendo de la época del año. Es muy importante tener presente que la efectividad de la acción de control está limitada por las posibilidades de refrigeración del actuador utilizado. En este caso un cooler de 12V para PC.

Dado que toda variación del setpoint implicaba un proceso lento de acceso y modificación de la memoria de programa e incluso en algunos casos de la base de reglas, se optó por realizar todas las pruebas con un setpoint fijo y se variaron las condiciones externas de operación.

Se graficaron los resultados obtenidos para diversas situaciones de operación, en cada figura aparece el comportamiento en lazo abierto comparado con el comportamiento en lazo cerrado. Todas las pruebas se realizaron con una temperatura ambiente de 24°C y el setpoint se fijó en 25°C

Situación I: Únicamente la carga 5 Conectada

Se ha llamado carga 5 a la última carga de derecha a izquierda la cual corresponde a una resistencia de 100 Ohmios y 5W, sobre esta se halla el sensor y también sobre esta actúa directamente el pequeño ventilador que proporciona la refrigeración forzada.

El sistema presenta una respuesta semejante a la de un sistema de primer orden, es evidente la mejoría con el controlador, luego de la estabilización de la temperatura la diferencia entre las dos respuestas esta alrededor de los 20 grados Celsius.

Situación II: Cargas 5 y 3 conectadas.

Al conectar las diferentes cargas las condiciones de corriente y de voltaje varían para el circuito debido a que la fuente de alimentación reacciona ante los cambios de carga dado que las resistencias son de bajo valor de ohmios y la regulación de las salidas no es optima en todo momento, esta situación permite tener momentos de operación diferentes que son útiles para observar el funcionamiento del controlador ante estos cambios.

En la grafica anterior se observa un cierto deterioro en la respuesta comparada con la situación anterior. También aparece un comportamiento oscilatorio. Debe tenerse en cuenta que la frecuencia de recolección de los datos fue de 30 segundos lo cual puede no ser óptimo y puede reflejarse en una baja calidad de la grafica ya que esta puede aparecer poco suavizada. Otro elemento a tener en cuenta es la posibilidad de que los puntos graficados hayan resultado insuficientes para captar el momento de estabilidad. En todo caso la mejora en el sistema esta vez es menor.

Situación III: Cargas 5,1 y 3 Conectadas.

En esta situación se nota de nuevo una respuesta favorable y el sistema logra tener de nuevo una diferencia que ronda los 20 grados entre la respuesta en lazo abierto y la respuesta en lazo cerrado aunque persisten las oscilaciones.

Situación IV: Todas las cargas conectadas.

Esta fue la última situación experimentada con cargas conectadas simultáneamente. Esta vez se conectaron todas las resistencias y se midió. De nuevo la grafica no aparece muy suave pero el efecto de la acción de control es satisfactorio.

Perturbaciones:

Para simular algunas perturbaciones sobre el sistema se procedió a realizar la conexión o desconexión de diversas cargas en momentos diferentes de operación para observar la reacción del controlador ya que en las cuatro situaciones anteriores las cargas estuvieron presentes simultáneamente con la carga objetivo de control.

Situación V: Cargas que se conectan o desconectan en 4 momentos diferentes.

Para esta ocasión se decidió ampliar un poco el número de muestras tomadas en busca de una mejor representación de las respuestas del sistema. Se observó este por un espacio de 20 minutos para lazo abierto y para lazo cerrado.

Se observa que en la gráfica quedan marcados con picos o depresiones los momentos en que se varía la situación de operación y en los que por lo tanto varía la respuesta del sistema ya sea en ausencia o en presencia del controlador.

Situación VI: Cargas que se conectan o desconectan en 10 momentos diferentes.

En esta ocasión la diferencia se nota bien marcada entre las dos situaciones bajo las cuales se analizó el sistema. La acción de control tiene un efecto de reducción de la temperatura en unos 20 grados aproximadamente. En contraposición con la gran cantidad de cambios sobre el sistema, la respuesta parece óptima, esto tal vez debido a que la base de reglas sobre las cuales se construyó el controlador cubre satisfactoriamente las condiciones de operación que aparecen bajo situaciones con una dinámica de cambio importante.

7. CONCLUSIONES

El control difuso ofrece nuevas posibilidades de trabajo en el campo de la automatización, este permite utilizar eficientemente recursos tecnológicos actuales como los microcontroladores y los programas de modelación y simulación de sistemas difusos, por lo tanto permite desarrollar aplicaciones que cubran requerimientos puntuales con especificaciones bien definidas.

La lógica difusa permite abordar los problemas de control de forma heurística y consiste en una opción muy interesante para superar dificultades grandes para la clasificación y modelación física y matemática de un sistema determinado dentro de los patrones tradicionales. La modelación lingüística es un medio rápido, flexible y efectivo que resulta adecuado para definir forma acertada el punto de partida del problema de control dentro de un escenario específico.

Los microcontroladores que el mercado actual ofrece disponen de una gran variedad de herramientas de desarrollo en el campo del software y del hardware que permiten que estos dispositivos acoplen muy bien dentro de los sistemas difusos de control. Las facilidades o prestaciones especiales de la gran mayoría de microcontroladores resultan singularmente adecuadas para el trabajo con control difuso.

La base de reglas así como el motor de inferencias de un control difuso pueden ser expresados o implementados con un grado muy alto de afinidad mediante las instrucciones o estructuras fundamentales de los lenguajes de alto nivel más populares.

Es necesario continuar desarrollando en nuestro entorno académico aplicaciones con

controladores difusos para disponer así de una base de conocimientos mas amplia que la actual, esto permitirá a los ingenieros de diferentes áreas y a las demás personas interesadas, poseer guías de trabajo dentro del campo del control difuso que provean observaciones que vayan mas allá de las premisas meramente teóricas o filosóficas del control difuso.

Los controladores difusos permiten ingresar a la base de reglas del sistema nueva información o los conocimientos valiosos de operarios y expertos de manera natural debido a la forma como se estructuran los sistemas difusos y a la componente *lingüística* y *subjetiva* de estos.

En general la filosofía del control difuso plantea procedimientos metódicos para desarrollar controladores y en ese aspecto es similar a las diversas corrientes del control tradicional. El proceso de diseño es sistemático y matemático, solo que el enfoque difuso posee la característica adicional de ser cercano al modo natural y lingüístico de los seres humanos de manejar los conocimientos y de realizar los procesos de análisis y toma de decisiones.

Existen procedimientos y métodos de análisis bien establecidos para ajustar o mejorar el comportamiento de los controladores difusos bajo condiciones especiales de operación o cuando las condiciones de diseño o de trabajo lo requieran.

Los controladores difusos pueden adaptarse con facilidad relativa a sistemas que posean, por su naturaleza, una dinámica especial de comportamiento y que por lo tanto presentan condiciones muy variables en cuanto a los puntos de operación.

Los microcontroladores de la gama alta e incluso los computadores personales ofrecen una alternativa de gran poder computacional y gran disponibilidad de recursos que pueden potenciar de manera excepcional el desarrollo de aplicaciones de control difuso de alta complejidad y altas prestaciones.

La aplicación desarrollada en este trabajo de grado puede mejorarse en el futuro agregándole funciones que le permitan al usuario un acceso fácil a funciones de ajuste de setpoint y cambio dinámico de la base de reglas.

BIBLIOGRAFIA

- [1] Geering. Hans P. Introduction to Fuzzy Control. Measurement and Control Laboratory Swiss Federal Institute of Technology. 1998.
- [2] Hao Yin, Dr. Bao-Gang Hug. Introduction to Fuzzy Control. International Journal of Fuzzy Systems, Vol. 5, No. 2, Junio 2003.
- [3] Muñoz, Carlos. Velásquez, Iván. Torres, Flavio. CONTROL DIFUSO PREALIMENTADO PARA UNA PLANTA PILOTO DE SECADO POR LECHO FLUIDIZADO Departamento de Ingeniería Eléctrica, Universidad de La Frontera, 2001.
- [4] Cilingiroglu, Ugur. Pamir, Banu. Gunay, Z. Sezgin. Dulger, Fikret. Sampled-Analog Implementation of Application-Specific Fuzzy Controllers IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 5, NO. 3, agosto 1997
- [5] Ying, Hao. A general technique for deriving analytical structure of fuzzy controllers using arbitrary trapezoidal input fuzzy sets and Zadeh and operator. Department of Electrical and Computer Engineering, Wayne State University, Detroit . Febrero 2003
- [6] Jantzen, Jan . Design Of Fuzzy Controllers. Technical University of Denmark, Department of Automation, Lyngby, DENMARK.Tech. report no 98-E 864 (design), 19 agosto 1998.
- [7] Domonkos, Tikk. On nowhere denseness of certain fuzzy controllers containing prerestricted number of rules. Technical University of Budapest Dept.of Telecommunications and Telematics, Hungría, 1997.

- [8] Galan, Peter. Temperature control based on traditional PID versus fuzzy controllers. Nortel Networks Control Software Design Documentation.
- [9] Gal A. Kaminka. On Problems of Knowledge in Fuzzy Control: Extended Abstract.
- [10] Egilegor, B. Uribe, J.P. Arregi, G. Pradilla, E. Susperreg, L. A FUZZY CONTROL ADAPTED BY A NEURAL NETWORK TO MAINTAIN A DWELLING WITHIN THERMAL COMFORT . IKERLAN Applied Research Centre.
- [11] Ponce. Sergio Damián. INCUBADORA DE CUIDADOS INTENSIVOS CONTROLADA CON LÓGICA DIFUSA Grupo de Ingeniería Clínica. UTN Facultad Regional San Nicolás.
- [12] Alvarez López, Ihosvany. Llanes Santiago, Orestes. Verdegay Galdeano, José Luis Una aplicación de control difuso para secado de tabaco en hojas Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada. Granada (España) Departamento de Automática y Computación. Facultad Eléctrica, ISPJAE. La Habana (Cuba)
- [13] Acosta, Gerardo. Taller de Controlador Basado en Lógica Difusa. Universidad Nacional del Centro de la Provincia de Buenos Aires. Noviembre 2001.
- [14] Ying, Hao. Chen, Guanrong. "Analytical Theory of Fuzzy Control with Applications" Information Science, Volumen 123, Abril 2000.
- [15] Ying, Hao. Fuzzy Control and Modeling. Analytical Foundations and Applications. IEEE Press. 2000
- [16] Binfet Jeremy. Wilamowski Bogdan M.. Microprocessor Implementation of Fuzzy Systems and Neural Networks. 2001. University of Idaho.
- [17] Kandel, Abraham. Langholz, Gideon. Fuzzy Hardware. Architectures and Applications. Kuwer Academic Publishers. 1998.
- [18] Lee, Ch. Ch.: "Fuzzy Logic Control Systems: Fuzzy Logic Controller – Part I and II", IEEE Trans. on SMC, Vol 20, N° 2, Marzo/Abril 1990, pp. 404-435.
- [19] Pedrycz, W.: "Fuzzy Control and Fuzzy Systems", Ed. John Wiley & Sons Inc., 1988.
- [20] Ibrahim, Ahmad M. Introduction to Applied Fuzzy Electronics. Prentice Hall, 1997.
- [21] National Semiconductors. LM35 Precision Centigrade Temperature Sensors. Data Sheet.
- [22] Microchip. PIC16F87X, 28/40-pin 8-Bit CMOS FLASH Microcontrollers. Data Sheet.
- [23] Leading Edge Technology Ltd . Manual de PIC Basic Compiler.