

51
2ej.



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

CONTROL PARA AUTOCLAVE IMPLEMENTADO
CON ALGORITMOS DE LOGICA DIFUSA
AUTOSINTONIZABLES.

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
(AREA ELECTRONICA)
P R E S E N T A N :
JOSE EMILIO DIAZ GONZALEZ Y
VICTOR MANUEL MARTINEZ REYES

DIRECTOR: ING. JESUS ALVAREZ CASTILLO



MEXICO, D. F.

1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**CONTROL PARA AUTOCLAVE
IMPLEMENTADO CON ALGORITMOS DE
LOGICA DIFUSA AUTOSINTONIZABLES**

*A mi Madre le doy las gracias por su
paciencia comprensión y cariño*

*A mis hermanos Néstor, Jorge, Blanca,
Julio y Dianita por su apoyo.*

*A todos mis profesores y a la Universidad
les doy las gracias por que de no ser por
ellos esto nunca hubiera podido ser*

*A todas las personas que de algún modo
han contribuido durante todos estos años a
mi desarrollo y formación profesional.
Para no olvidar a nadie solo quiero decir
GRACIAS por todo.*

Victor Manuel Martínez Reyes.

À ma maestra, harmonica
y
à la memoria de mi querida patria.

José Emilio Díaz Banzález.

INDICE

INDICE.

	Pag.
CAPITULO 1 INTRODUCCION.	1.
CAPITULO 2 FUNDAMENTOS DE LOGICA DIFUSA.	4.
2.1 Introducción.	4.
2.2 Descripción de la lógica difusa.	4.
2.2.1 Antecedentes de la lógica difusa	4.
2.2.2 Características principales de la lógica difusa.	5.
2.2.3 Aplicaciones de la lógica difusa.	6.
2.3 Fundamentos matemáticos de la lógica difusa.	6.
2.3.1 Los conjuntos convencionales.	6.
2.3.2 Conjuntos difusos.	9.
2.3.2.1 Representación gráfica de los conjuntos difusos.	9.
2.3.2.2 Características de los conjuntos difusos.	9.
2.3.2.3 Elementos de una función miembro.	13.
2.4 Variables lingüísticas y razonamiento aproximado.	14.
2.5 Estructura de un algoritmo basado en lógica difusa.	17.
2.6 Parámetros de diseño de un algoritmo basado en lógica difusa	19.
CAPITULO 3 ESTRUCTURA ALGORITMICA DEL CONTROLADOR DIFUSO AUTOSINTONIZABLE	25.
3.1 Introducción.	25.
3.2 Características y clasificaciones de los algoritmos de control.	25.
3.2.1 Características de los algoritmos de control.	26.
3.3 Descripción general de la estructura del algoritmo de control utilizado.	27.
3.3.1 Características principales de cada etapa del proceso.	34.
3.3.2 Descripción general del proceso.	34.
3.4 Algoritmo difuso utilizado en el control del proceso.	34.
3.4.1 Proceso de fusificación.	36.
3.4.2 Generación de reglas de inferencia.	37.

3.4.3	Determinación de las alturas en base a la fusificación de las entradas.	39.
3.4.4	Determinación del valor numérico de la salida en base a la defusificación de las alturas de las entradas.	41.
3.4.5	Estabilidad del controlador difuso.	41.
3.4.6	Autosintonización del controlador difuso.	41.
CAPITULO 4. DISEÑO DE LA ARQUITECTURA DEL SISTEMA		42.
4.1	Introducción.	42.
4.2	Procesamiento paralelo	42.
4.2.1	Paralelismo	42.
4.2.2	Características del procesamiento paralelo	43.
4.3	Clasificación del procesamiento paralelo	43.
4.3.1	Por su estructura	43.
4.3.2	Por su algoritmo	47.
4.3.2.1	Clasificación de Flynn	47.
4.3.2.1.1	Multiplicidad de flujos de instrucciones y datos	47.
4.3.3	Clasificación de Feng	50.
4.3.3.1	Procesamiento serie frente a procesamiento paralelo	50.
4.3.4	Clasificación de Handler	52.
4.3.4.1	Paralelismo frente a segmentación encauzada	52.
4.4	Características de la segmentación encauzada	53.
4.4.1	Clasificación de procesadores de segmentación encauzada.	55.
4.5	¿Porqué segmentación encauzada? (PIPELINE)	59.
4.6	Arquitectura del controlador	60.
4.7	Microprocesadores y microcontroladores	61.
4.7.1	Estructura interna	62.
4.7.2	Tipos y características	63.
4.7.2.1	Características	64.
4.7.2.2	Características eléctricas	65.
4.7.3	Criterios de selección	65.
4.7.4	Dispositivos (microprocesadores) seleccionados	66.
4.8	Diagrama a bloques de la arquitectura	68.
4.9	Protocolos	69.

CAPITULO 5. TRANSDUCTORES Y SENSORES	71.
5.1 Introducción.	71.
5.2 Transductores de entrada y sensores.	71.
5.3 Transductores y sensores de temperatura.	72.
5.3.1 Detectores de resistencia de temperatura (RTD's).	72.
5.3.2 El termopar.	74.
5.3.2.1 Fuentes de error de los termopares.	75.
5.4 Termistores.	76.
5.4.1 Características de resistencia - temperatura de termistores y RTD's.	77.
5.4.2 Técnicas de fabricación de los termistores.	77.
5.4.3 Tipos de termistores existentes.	79.
5.4.4 Pruebas y calibración de termistores.	79.
5.4.5 Retraso en la respuesta o retardo.	80.
5.4.6 Estabilidad de termistores.	81.
5.4.7 Linealidad de termistores.	81.
5.5 Transductores y sensores de presión.	82.
5.5.1 Sensores de presión absoluta.	83.
5.5.2 Sensores de presión Gage.	83.
5.5.3 Sensores de presión diferencial.	83.
5.5.4 Transductor de presión inductiva.	83.
5.5.5 Transductores de presión piezoeléctricos.	85.
5.6 Consideraciones.	85.
CAPITULO 6. PRUEBAS Y RESULTADOS	86.
6.1 Introducción.	86.
6.2 Pruebas.	86.
6.2.1 Algoritmos.	86.
6.2.2 Arquitectura.	87.
6.3 Resultados.	88.
6.4 Manejo del proceso desde el monitor central.	89.
CAPITULO 7. CONCLUSIONES	91.

ANEXOS

**ANEXO 1. Diagrama de tiempos para cada microprocesador
y lista de tareas ejecutadas por paso**

94.

95.

ANEXO 2. Características eléctricas de cada microprocesador

97.

CAPITULO. 1

INTRODUCCION.

El presente proyecto trata de una arquitectura a base de tres microprocesadores que forman un controlador difuso autosintonizable y su función es la de controlar la presión y la temperatura dentro de una autoclave. Los microprocesadores usados son: el 68HC11F1, 68HC11A8 así como también el 8088 de Intel. Esto es con la finalidad de usar dispositivos baratos y poderosos.

El proceso se puede dividir en los siguientes pasos:

- Lectura del valor que tienen las variables de entrada (P_1 , T_1 , P_2 , T_2)
- Fusificación de las entradas
- Evaluación de reglas.
- Defusificación
- Autosintonización y ejecución (manipulación de actuadores)
- Salidas

Las variables de entrada P_1 , T_1 , P_2 , T_2 entran directamente al 68HC11F1 que es en este caso el microcontrolador que va a fusificar. Por medio del puerto A para las direcciones y el puerto B para los datos se conecta el 68HC11F1 con el 68HC11A8. Este último microcontrolador realiza la función de defusificación.

Después de realizada la función de defusificación, el siguiente paso es la autosintonización y ejecución que se va lograr conectando como la última etapa y a través de un latch (74LS373) el microprocesador 8088. Se han seguido estos pasos de fusificación del proceso de acuerdo a los pasos del método de Von Neumann que consisten en:

- Entradas
- Buscar
- Codificar
- Ejecutar
- Salidas

La lógica difusa es un método que fácilmente representa procesos analógicos sobre una computadora digital.

Una variable de temperatura en lógica difusa puede tener un rango de estados como: frío, semifrío, moderado, tibio, caliente y muy caliente, el cambio de un estado a otro no está bien definido y las declaraciones de control son escritas en términos de estas ideas imprecisas que

constituyen los estados de la variable. Como ejemplo de control difuso tenemos el sistema de frenos de un automóvil.

Si la temperatura del freno es "tibia" y la velocidad "no es muy rápida", entonces la presión del freno es "ligeramente decrementada".

Ahora tenemos el mismo sistema analizado desde el punto de vista de un control "Proporcional Integral y Derivativo"(PID). Si la temperatura del freno es mayor de 280°C y la velocidad es menor de 45 km/h, entonces la presión del freno es 190 pascales. Como podemos darnos cuenta este último análisis utilizando el sistema PID es más específico

A continuación enlistaremos los pasos que conforman a un sistema difuso típico:

- Entrada
- Fusificador
- Proceso lógico
- Defusificador
- Salida

La entrada es leída desde una fuente externa y es adaptada para ser procesada por la lógica difusa (fusificada). La salida del proceso lógico es defusificada antes de ser enviada al sistema físico externo a controlar.

Un sistema de control convencional tiene toda la estructura muy similar pero sin los elementos difusos y es como se ilustra a continuación

- Entrada.
- Proceso lógico.
- Salida.
- El núcleo de este proceso

El centro de la diferencia entre la lógica clásica y la difusa está en la llamada ley del medio excluido. Las ciencias de la computación tratan de modelar problemas complejos del mundo real con un número infinito de variables con simples valores bivalentes. Esto es, verdadero o falso o sea uno o cero, pero la región transitoria se pierde. Con la lógica multivalente o lógica Difusa podemos modelar tales transiciones vagas. Podemos comparar la variedad de tonos del color verde. Podemos ver y utilizar el área gris que existe entre el negro y el blanco. Estas áreas grises pueden entonces ser comparadas de tal manera que una decisión humana puede ser simulada.

Esto es el porqué la lógica difusa es usada en control y/o en situaciones de toma de decisiones. Podemos modelar con mayor precisión el comportamiento del mundo real con menos complicación. Esto simplifica el diseño, minimiza el costo y mejora el desempeño.

A continuación describiremos el contenido de nuestro trabajo. En el capítulo uno se da una descripción de manera general acerca de las ventajas y desventajas del método utilizado (lógica difusa) para el desarrollo del proyecto. Del mismo modo se dan a continuación las características que tiene de acuerdo a su organización estructural y las razones por las cuales se hicieron de esta manera.

El capítulo dos nos da un panorama histórico de la lógica difusa así como también los fundamentos matemáticos en los que se basan los algoritmos de lógica difusa. Tenemos una comparación de los controles convencionales con respecto de la lógica difusa y sus aplicaciones mencionando sus ventajas, también se mencionan las variables lingüísticas y su utilización. Como veremos más adelante este método se está aplicando en muchas áreas del quehacer humano así como en otras está por aplicarse.

En el capítulo tres tenemos las características y clasificaciones de los algoritmos de control, mencionando los tipos de este, así como también los casos discretos o continuos, de lazo abierto o de lazo cerrado, etc. Se da la descripción general de la estructura del algoritmo de control que se usó en este proyecto, que es del tipo *PIPELINE*. También aparecen las características principales de cada etapa del proceso así como la fusificación y la generación de las reglas de inferencia. Después tenemos la determinación del valor numérico de la salida en base a la fusificación de las alturas de las entradas. La estabilidad del controlador difuso y la autosintonización.

En el capítulo cuatro se ve la programación de los algoritmos de control difuso que se basa en las reglas de inferencia las cuales permiten controlar el proceso sin tener que usar un modelo matemático. Se habla también de los protocolos, el tipo y cuantos bauds maneja, la estructura y el formato, se menciona también la descripción de la programación a manera de bloques. Por otro lado este capítulo también hace mención de la arquitectura: el porque de este diseño, las partes que la componen y el tipo de procesamiento *PIPELINE*.

En el capítulo cinco damos una definición de lo que son los transductores y también de lo que es un sensor, también un panorama de lo que es su funcionamiento. Se da la variedad de lo que hay en el mercado. Se habla acerca de los termistores que son y como funcionan y los tipos que existen así como también los materiales con que están hechos y las técnicas de fabricación.

En el capítulo seis se muestra a través de los resultados obtenidos durante la etapa de pruebas lo confiable que son los algoritmos de control difuso, así como las pruebas realizadas al prototipo resultado del proyecto.

El capítulo siguiente que es el siete trata de las conclusiones a las que llegamos después de terminado el proyecto.

CAPITULO 2

FUNDAMENTOS DE LOGICA DIFUSA

2.1 INTRODUCCION

Con respecto a este capítulo, debemos considerar que tipo de procesos vamos a controlar de tal forma que implementemos los algoritmos de control adecuados, o sea los que nos permitan obtener resultados con una buena precisión y en el menor tiempo de cálculo posible, así como el uso de una mínima cantidad de memoria; además debemos tomar en cuenta que su implementación deberá ser de bajo costo.

Existe un tipo de algoritmo de control el cual veremos con mayor detalle en el siguiente capítulo, que basa su estructura en la teoría de lógica difusa. La cual es un método que nos facilita la implementación de sistemas de control, esto debido a su facilidad de manejo e.

En este capítulo mencionaremos las características más importantes de la lógica difusa, se dará una breve semblanza de esta, plantearémos las bases matemáticas para su justificación, así como sus ventajas y desventajas. Para que al final estemos en la posibilidad de escoger el algoritmo (basado en lógica difusa) más adecuado a las características de nuestro proceso. Y dar paso para que en el siguiente capítulo se realice el análisis en forma detallada del algoritmo planteado para el desarrollo del sistema.

2.2 DESCRIPCION DE LA LOGICA DIFUSA.

2.2.1 ANTECEDENTES DE LÓGICA DIFUSA

La lógica difusa (Fuzzy. Logic), nace en la Universidad de California, en Berkeley, Estados Unidos, en el año de 1965 [1]. Fue creada por el profesor Lofti A. Zadeh, aparece después de la lógica multinivel la cual propone tres niveles lógicos: VERDADERO (1); FALSO (0) y NEUTRO ($\frac{1}{2}$), el cual representa la mitad verdadera y la mitad falsa. Fue propuesta por Lukasiewicz en 1930 [1]. La Lógica Difusa a diferencia de la anterior abarca una infinidad de valores los cuales están comprendidos entre 0 (completamente falso) y 1 (completamente verdadero). Tiene aplicaciones tanto en sistemas expertos, sistemas de control, etc., siendo sus aplicaciones principales el área numérica, y la Ingeniería de conocimiento. Aunque existen otras como la primera aplicación industrial en 1977, que se realizó en la industria del cemento en Dinamarca [16].

EN TÉRMINOS GENERALES LOS SISTEMAS DIFUSOS SON CONVENIENTES CUANDO EXISTE INCERTIDUMBRE O RAZONAMIENTO APROXIMADO. Es decir cuando los modelos matemáticos que describen el comportamiento del sistema son demasiado complejos y su implementación en una arquitectura como la que proponemos para nuestro sistema es muy complicada [15].

En otras palabras, podemos decir que los métodos convencionales son buenos para resolver problemas simples en cuanto a su modelado, mientras que los sistemas basados en lógica difusa son convenientes para problemas complejos o aplicaciones que involucren descripciones humanas o pensamientos intuitivos.

2.2.2 CARACTERÍSTICAS PRINCIPALES DE LA LÓGICA DIFUSA.

Dentro de los principales parámetros que caracterizan a la lógica difusa podemos mencionar los siguientes:

- **CAPACIDAD DE APRENDIZAJE:** Donde se añaden ciertos datos para configurar un mejor control.
- **REGLAS DE INFERENCIA:** Las cuales, están diseñadas en base a un determinado conocimiento del problema y a la forma en que un experto lo resolvería.

En la Lógica Difusa se manejan conceptos ambigüos como caliente, muy caliente, viejo, joven etc., que reciben el nombre de "variables lingüísticas". En esta lógica lo importante es el grado de ocurrencia, sin importar el paso de tiempo o el número de pruebas realizadas, como se tendría que hacer en los casos siguientes; "estará nublado mañana", o "al aventar un dado en la prueba x dará un número n ".

La teoría de la Probabilidad, está basada en el paso de tiempo o en la realización de *n pruebas*, por lo que podemos decir que mide la probabilidad de ocurrencia o de no ocurrencia de un evento, en tanto que la teoría de lógica difusa mide el grado de ocurrencia de un evento o de una determinada condición. De lo anterior se puede decir que la Probabilidad se basa en razonamientos exactos, mientras que la Lógica Difusa está basada en razonamientos aproximados, por lo tanto, para desarrollar un sistema difuso, podemos diseñar un método que cumpla los siguientes pasos [2], [3], [15] :

1. Determinar si la estructura del problema a resolver amerita el uso de lógica difusa.
2. Si es así, determinar los rangos de las variables de entrada y de salida.
3. Definir las funciones que constituyen las salidas y las entradas, siendo estas las posibles opciones que tomarán las entradas y salidas.
4. Construir las reglas que relacionarán las entradas y salidas. (se requiere de pruebas exhaustivas para verificar su correcto funcionamiento)

2.2.3 APLICACIONES DE LA LÓGICA DIFUSA.

En términos generales la lógica difusa se aplica en muchas áreas y esta por aplicarse en otras. Dichas aplicaciones son tan amplias que pueden ir desde comprensión de lenguaje, interpretación de información humana, relaciones humanas, planeación, toma de decisiones hasta control de robots, reconocimiento de patrones, etc [16].

2.3 FUNDAMENTOS MATEMÁTICOS DE LA LÓGICA DIFUSA

2.3.1 LOS CONJUNTOS CONVENCIONALES.

Con respecto a los conjuntos convencionales se puede mencionar que los valores o están dentro del conjunto o no, de acuerdo a la siguiente ecuación [1].

$$X(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad \text{ec. 2.1}$$

Por ejemplo si hablamos de un conjunto que caracterice la edad, bajo este esquema el conjunto puede ser dividido en 3 subconjuntos :

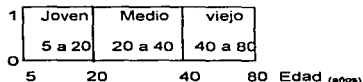


Fig. 2.1 Conjuntos Crips

A dichas divisiones se les llamará conjuntos crips. Cada conjunto crips contiene un determinado rango de valores, existiendo un solo valor en común entre ellos, al cual se le pueden aplicar las operaciones básicas de conjuntos, para cada una de ellas se tendrá su representación gráfica, que en forma general serán las siguientes (solo mencionaremos las más importantes para nuestro trabajo) [1] :

Unión: $\{x / x \in A \text{ ó } x \in B\} = A \cup B$

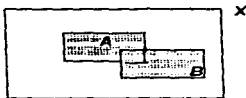


Fig. 2.2 Unión

Intersección: $\{x / x \in A \text{ y } x \in B\} = A \cap B$

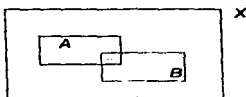


Fig. 2.3 Intersección

Complemento: $\{x / x \notin A, x \in X\} = \bar{A}$

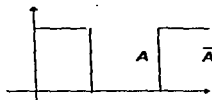


Fig. 2.4 Complemento

Donde se tienen las siguientes propiedades:

- Comutabilidad: $A \cup B = B \cup A$ ó $A \cap B = B \cap A$ **ecs.2.2**
- Asociatividad: $A \cup (B \cup C) = (A \cup B) \cup C$ ó $A \cap (B \cap C) = (A \cap B) \cap C$ **ecs.2.3**
- Distributividad: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ ó $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ **ecs.2.4**
- Idempotencia: $A \cup A = A$ ó $A \cap A = A$ **ecs.2.5**
- Identidad: $A \cup \emptyset = A$; $A \cap X = A$; $A \cap \emptyset = \emptyset$; $A \cup X = X$ **ecs.2.6**
- Transitividad: Si $A \subseteq B \subseteq C$, entonces $A \subseteq C$ **ecs.2.7**
- Involución: $\overline{\overline{A}} = A$ **ecs.2.8**

Leyes de Morgan: $\overline{(A \cap B)} = \overline{A} \cup \overline{B} = \{x / x \in A \text{ or } x \in B\}$

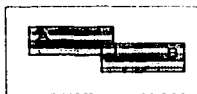


Fig. 2.5 Intersección

$$\overline{(A \cup B)} = \overline{A} \cap \overline{B} = \{x / x \in A \text{ y } x \notin B\}$$

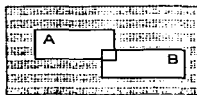


Fig. 2.6 Unión complemento

2.3.2 CONJUNTOS DIFUSOS.

Su representación esta dada por medio de la siguiente expresión, para la que existen una infinidad de valores dentro del intervalo de 0 a 1 [1].

$$\mu(x) \in [0,1] \quad \text{ec. 2.9}$$

2.3.2.1 REPRESENTACIÓN GRÁFICA DE LOS CONJUNTOS DIFUSOS.

Existen varias formas de representar a los conjuntos difusos. Puede ser como trapezios, triángulos e incluso en forma de curvas [15].

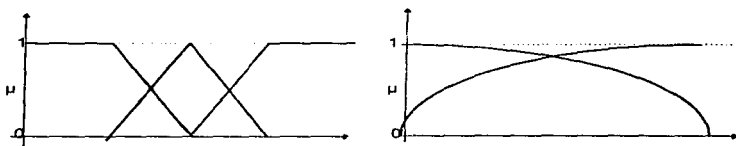


Fig. 2.7 Representación gráfica de los conjuntos difusos

2.3.2.2 CARACTERÍSTICAS DE LOS CONJUNTOS DIFUSOS.

Los conjuntos difusos cuentan con los siguientes elementos [1] :

- **Función de pertenencia:** Es el grado en el cual el elemento $\mu(x)$ pertenece al subconjunto A

$$\mu_A(x) = \text{grado}(x \in A). \quad \text{ec. 2.10}$$

- **Conjunto (set):** X es el universo, representa a todos los valores posibles del sistema.

-Subconjunto (subset) A: Son valores que se encuentran dentro del conjunto X y ϵA

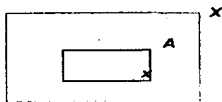


Fig. 2.8 Representación de un Subconjunto

Donde $\mu_x \rightarrow [0,1]$ podrá tener un número infinito de valores entre 0 y 1

Si hablamos de divisiones que caracterizan un conjunto difuso por ejemplo la edad, como en el caso anterior, el conjunto edad puede ser dividido en 3 subconjuntos, de forma triangular, trapecoide o bien de campana (mostrando la trapecoide y la triangular)

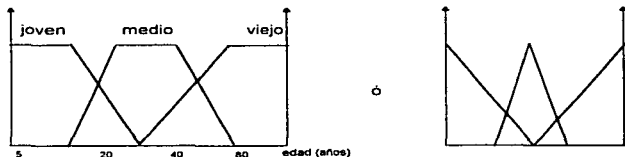


Fig. 2.9 Representación de los conjuntos difusos

A dichas divisiones se les llamara conjuntos difusos. Para ejemplificar la relación de estos conjuntos propondremos los conjuntos A, B y C, cuyas operaciones básicas estarán dadas de la siguiente forma [2], [7] :

Unión:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$$

ec. 2.11

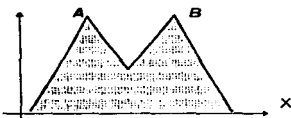


Fig. 2.10 Unión entre conjuntos difusos

Intersección:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$$

ec. 2.12

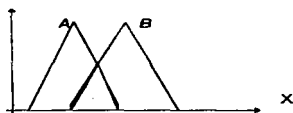


Fig. 2.11 Intersección entre conjuntos difusos.

Complemento:

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x)$$

ec. 2.13

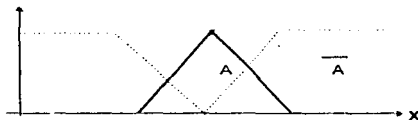


Fig. 2.12 Complemento de conjuntos difusos

Donde se tienen las siguientes propiedades:

Conmutatividad:

$$\begin{aligned} \mu_{A \cup B}(x) &= \mu_A(x) \vee \mu_B(x) = \mu_{B \cup A}(x) = \mu_B(x) \vee \mu_A(x) \\ \mu_{A \cap B}(x) &= \mu_A(x) \wedge \mu_B(x) = \mu_{B \cap A}(x) = \mu_B(x) \wedge \mu_A(x). \end{aligned} \quad \text{ecs. 2.14}$$

Las propiedades de Asociatividad, Distributividad, Idempotencia, Identidad, Transitiividad e Involución son exactamente igual para los Conjuntos crisp que para los difusos.

Ley de Morgan: $\mu_{\overline{A \cap B}}(x) = \mu_{\overline{A} \cup \overline{B}}(x) = \mu_{\overline{A}}(x) \vee \mu_{\overline{B}}(x).$ ec. 2.15

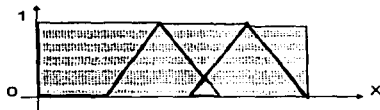


Fig. 2.13 Intersección de conjuntos

$$\mu_{\overline{A \cup B}}(x) = \mu_{\overline{A} \cap \overline{B}}(x) = \mu_{\overline{A}}(x) \wedge \mu_{\overline{B}}(x). \quad \text{ec. 2.16}$$

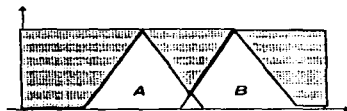


Fig. 2.14 Unión de conjuntos difusos

Definiendo

$$\begin{aligned} \mu_{A \cup B}(x) &= \mu_A(x) \vee \mu_B(x) = \max[\mu_A(x), \mu_B(x)] \\ \mu_{A \cap B}(x) &= \mu_A(x) \wedge \mu_B(x) = \min[\mu_A(x), \mu_B(x)] \end{aligned} \quad \text{ecs. 2.17}$$

2.3.2.3.- ELEMENTOS DE UNA FUNCIÓN MIEMBRO

De acuerdo a la siguiente figura se tendrán los siguientes elementos [1]:

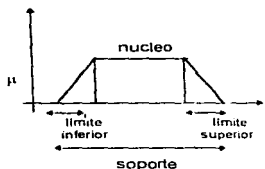


Fig. 2.15 Elementos de una función miembro

Donde un subconjunto A del conjunto X estará definido por la siguiente ecuación:

$$A = \mu_A(x_1) / X_1 + \dots + \mu_A(x_n) / X_n = \sum \mu_A(x_i) / X_i = \int_x \mu_A(x) / X \quad \text{ec. 2.18}$$

Aquí los elementos de soporte están en el numerador y el grado en el denominador. Usando esta última ecuación podemos expresar la unión, la intersección y el complemento de la siguiente forma [1]:

$$A \cup B = \int_x [(\mu_A(x) \vee \mu_B(x)) / X] \quad \text{ec. 2.19}$$

$$A \cap B = \int_x [\mu_A(x) \wedge \mu_B(x)] / X \quad \text{ec. 2.20}$$

$$\bar{A} = \int_x [1 - \mu_A(x)] / X \quad \text{ec. 2.21}$$

2.4 VARIABLES LINGÜÍSTICAS Y RAZONAMIENTO APROXIMADO

Una variable lingüística es el nombre que se le da a la característica que representa el conocimiento sobre "algo", y no es otra cosa que un subconjunto del conjunto universal X , donde a dicha variable se le podrán asignar ciertos atributos, que no serán otra cosa que los conjuntos difusos.

Una variable lingüística se puede representar con una función de 5 elementos $\{X, T(X), U, G, M\}$, donde X es el nombre de la variable; $T(X)$, es el conjunto de nombres de valores lingüísticos o características, que describen el conocimiento de "algo", siendo cada uno de ellos un conjunto difuso, definido en el universo U . G es una regla sintáctica para generar nombres de valores de X y M es una regla semántica para asociar con cada valor de X el significado. De tal forma que $M(X)$ denota un subconjunto de U . Por ejemplo velocidad es interpretada como una variable lingüística, tal que $T(\text{velocidad}) = \{\text{despacio, moderado, rápido, muy despacio etc}\}$, donde están los conjuntos difusos relacionados con esta variable y el universo estará definido por $U = [0, 100]$, definiendo los límites para cada conjunto difuso como sigue [3]:

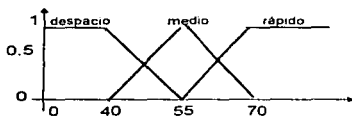


Fig. 2.16 Ejemplo de variables lingüísticas

Otros ejemplos pueden ser:

$T(\text{edad}) = \{\text{joven} + \text{no joven} + \text{muy joven} + \text{no muy joven} + \text{muy muy joven} + \dots + \text{viejo} + \text{no viejo} + \text{muy viejo} + \text{no muy viejo} + \dots + \text{no muy joven "y" no muy viejo} + \dots + \text{edad media} + \text{no edad media} + \dots + \text{no viejo} + \text{no edad media} + \dots + \text{extremadamente viejo} + \text{viejo "o" joven} + \text{etc.}\}$

$T(\text{apariencia}) = \{\text{bonita} + \text{atractiva} + \text{fea} + \dots + \text{más o menos bonita} + \text{horrible} + \dots + \text{etc.}\}$

Por otra parte es conveniente ejemplificar el concepto de "característica" mencionado anteriormente. Esto puede hacerse de la siguiente forma; supongamos que hablamos de las

características más comunes del hombre las cuales quedarán conformadas de la siguiente forma:

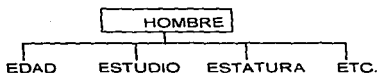


Fig. 2.17 Conjuntos de características

Dichas características serán los conjuntos difusos del conjunto hombre. Por otra parte la estructura jerárquica de una variable lingüística indica que una subcaracterística puede pertenecer a más de una característica fig. 2.18 y 2.19

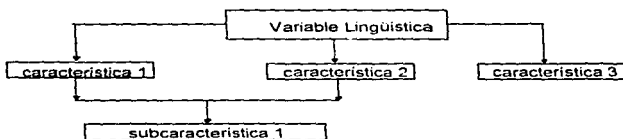


fig. 2.18 Estructura jerárquica de las variables lingüísticas

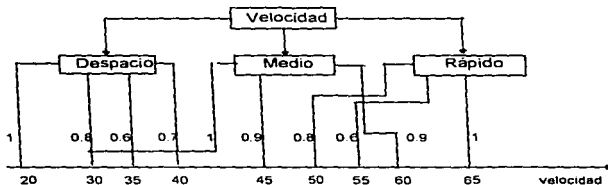


Fig.2.19 Ejemplo de las características de la velocidad

Donde podemos apreciar que existen valores de velocidad que pertenecen a más de un subconjunto difuso con diferente grado de pertenencia para cada uno de ellos (funciones miembro). Con referencia a esto la función de compatibilidad asigna a cada valor de la variable, uno en el intervalo $[0, 1]$, que es el rango de la función miembro, por ejemplo las de la siguiente figura [3]:



Fig. 2.20 Ejemplo de funciones miembro.

Cada conjunto difuso tiene un rango en X . Si tenemos un conjunto difuso A , y una μ_A asociada a cada elemento x de U , un número $\mu_A(x)$ en el intervalo $[0, 1]$, representará el grado de pertenencia de x en A . Donde como se había definido:

$$A = \int_U \mu_A(y)/y. \quad \text{ec. 2.22}$$

$$A = \mu_1 / y_1 + \dots + \mu_n / y_n; \quad A = \sum_{i=1}^n \mu_i / y_i \quad \text{ec. 2.23}$$

en donde μ_i para toda $i = 1, 2, 3, \dots, n$; es el grado de pertenencia de x_i en A , en

$$\text{donde } U = x_1 + x_2 + \dots + x_n \quad \text{ó} \quad U = \sum_{i=1}^n x_i \quad \text{ec. 2.24}$$

$$\text{o también } U = 1/x_1 + 1/x_2 + \dots + 1/x_n \quad \text{ó} \quad U = \sum_{i=1}^n 1/x_i \quad \text{ec. 2.25}$$

De este modo para los números reales a y b podemos definir,

$$a \vee b = \text{máx} (a, b) = \begin{cases} a, & \text{si } a \geq b \\ b, & \text{si } a < b \end{cases} \quad \text{ec. 2.26}$$

$$a \wedge b = \text{mín} (a, b) = \begin{cases} a, & \text{si } a \leq b \\ b, & \text{si } a > b \end{cases} \quad \text{ec. 2.27}$$

Por lo tanto las relaciones de conjunto difusos simples (no pares ordenados) quedarían:

Unión: $A \cup B = \int_U [\mu_A(y) \vee \mu_B(y)] / y$, de este modo u "o" $v = u \cup v$ ec. 2.28

Intersección: $A \cap B = \int_U [\mu_A(y) \wedge \mu_B(y)] / y$, de este modo u "y" $v = u \cap v$ ec. 2.29

Complemento: $A = \int_U [1 - \mu_A(y)] / y$ ec. 2.30

Producto: $AB = \int_U \mu_A(y) \mu_B(y) / y$. ec. 2.31

Multipliación por un número: $\alpha A = \int_U \alpha \mu_A(y) / y$. ec. 2.32

2.5 ESTRUCTURA DE UN ALGORITMO BASADO EN LÓGICA DIFUSA

La estructura básica de un algoritmo basado en Lógica Difusa está constituido por 4 bloques principales los cuales están relacionados entre sí de tal manera que operan de forma sistemática para desempeñar una determinada función. Dicha estructura está conformada de la siguiente manera [16]:

1) **FUSIFICACION:** La cual desempeña las siguientes funciones:

- Mide los valores de las variables de entrada
- Ubica en la escala de valores del correspondiente universo a dichas variables.
- Convierte los datos de entrada en un grado de pertenencia de uno o más conjunto difusos

2) **BASE DE CONOCIMIENTOS:** Contiene las reglas en las cuales se basa el sistema para desempeñar su trabajo. Se constituye de premisas, ubicando los conjuntos difusos de las variables involucradas en cierto arreglo que le permita al bloque siguiente usar dichas reglas.

3) **BLOQUE DE TOMA DE DECISIONES:** Este bloque y el anterior se usan para simular la experiencia, intuición y aprendizaje de un operador humano (experto), el cual sustituirá el modelo o modelos matemáticos, que nos permiten tener control del sistema. Este bloque está basado en reglas de inferencia, que permiten tomar una decisión para el control de una acción en base a los conjunto difusos de las variables involucradas, construidos a partir de la experiencia del experto. En forma genérica se tendrán reglas del siguiente tipo:

Si un conjunto de condiciones son satisfechas entonces un conjunto de consecuencias pueden ser inferidas. Estas reglas pueden tomar la siguiente forma que podrá ser modificada de acuerdo a cada sistema.

R Si X es A y Y es B entonces Z es C,
 R Si X es A y Y es B entonces Z es C,
 R Si X es A y Y es B entonces Z es C,

4) **DEFUSIFICACIÓN:** Desempeña las siguientes funciones:

- Se encarga de tomar el conjunto de valores fusificados para procesar estos mediante las reglas de inferencia establecidas y dar un conjunto de valores a las consecuencias inferidas.
- Dicho conjunto de valores se mapean, de tal forma que indiquen a que subconjunto difuso de las variables de salida pertenecen y dar un determinado valor como resultado del proceso. En forma de diagramas de bloques el sistema quedará de la siguiente manera [3]

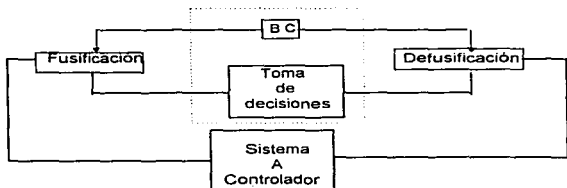


Fig.2.21 Estructura de un algoritmo basado en lógica difusa.

2.6 PARAMETROS DE DISEÑO DE UN ALGORITMO BASADO EN LÓGICA DIFUSA

Los parámetros que se tienen para el diseño de un algoritmo basado en Lógica Difusa son una característica importante, ya que cada algoritmo tiene un número y tipo diferente de variables de entrada y salida, a continuación mencionaremos dichos parámetros dando una breve explicación de su o sus objetivos [7], [8].

1) **ESTRATEGIAS DE FUSIFICACIÓN:** Como lo habíamos mencionado anteriormente la fusificación consiste en convertir un conjunto crisp en un conjunto difuso. Para poder representar los conjuntos difusos existen básicamente 3 tipos de funciones las cuales permiten describir la variable lingüística en términos de valores, estas funciones dependen de que tipo de universo se tenga (continuo o discreto), y son las siguientes:

- **Definición Funcional:** Esta definición es para un universo de tipo continuo, dicha definición permite conocer todas las funciones miembro involucradas en el subconjunto correspondiente, matemáticamente cada función miembro estará representada por:

$$\mu_i(x) = C \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right) \quad \text{ec. 2.33}$$

donde μ_i y σ_i son parámetros de normalización que sirven para determinar las funciones miembro [1].

- **Definición numérica:** Este tipo de definición se utiliza para universos discretos, donde los conjuntos difusos están representados por trapecios o triángulos y además matemáticamente las funciones miembro estarán representadas por

$$\mu_i(u) = \sum a_i / \mu_i \quad \text{para } i = 1, \dots, n \quad \text{ec. 2.34}$$

donde μ_i son los posibles valores entre 0 y 1 que se tienen para cada uno de los subconjuntos. Estos subconjuntos tendrán las siguientes formas,

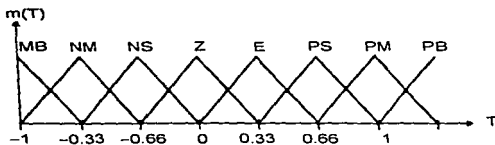


Fig. 2.22 Representación gráfica de la representación numérica

De esta forma podemos representar conjunto difusos de manera general como:

$$A(x) = \begin{cases} \phi_i(x), & x \in [P_0, P_1] \\ \phi_i(x), & x \in [P_{i-1}, P_i] \end{cases}$$

donde $\phi_i(x) \in [0,1]$ para $x \in [P_{i-1}, P_i]$; $i = 1,2,\dots,s$; $y-w = P_0 < P_1 < \dots < P_{s-1} < P_s = w$
 ec. 2.35

$\phi_i(x) = \sum_{k=0}^n C_k^i X^k$ donde C_k^i son parámetros del polinomio $\phi_i(x)$. ec. 2.36

Como ejemplo tomamos dos conjunto difusos, los cuales estarán representados por la siguientes figuras:

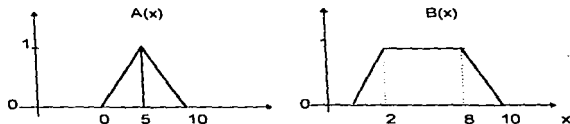


Fig. 2.23 Conjuntos difusos

De esto podemos concluir que el producto de dos funciones miembro continuas da como resultado una función miembro continua [17].

2) **BASES DE DATOS:** En este punto tomaremos en cuenta los siguientes aspectos:

- Discretización: Donde asignamos valores numéricos a cada función miembro, de acuerdo al valor de la variable tratada.
- Se colocan los subconjuntos difusos tomando en cuenta el rango de cada uno.
- Tipos de conjuntos difusos

3) **IMPLEMENTACIÓN DE REGLAS**

- Selección de Entradas y Salidas para construir las matrices de reglas.
- Interpretación de reglas de conectividad: (si entonces, no, "y", "o", entonces muy, más o menos etc.)

En la figura siguiente se puede apreciar lo que se refiere a este punto:

	Frio	Fres	Tibio	Cal	M. Cal.
Deb			PM		
Baja			PS		
Corr	PS	PS	Z	NS	NL
Alta			NS		
M. Alt			NM		

Fig. 2.24 Matriz de implementación de reglas

Esta matriz contiene las posibles regiones que toman las variables de entrada, así como los posibles valores que puede tomar la variable de salida, los cuales se encuentran localizados en los puntos de intersección en la matriz. Las entradas serán la presión cuyas regiones son:

Débil, Baja, Correcta, Alta y Muy Alta. Y la temperatura cuyas regiones son las siguientes: Frio, Fresco, Tibio, Cálido y Caliente. Lo que dará como salida un determinado ángulo de apertura de una válvula, el cual podrá estar en las siguientes regiones: PS (Positivo Pequeño), PM (Positivo Mediano), NS (Negativo Pequeño), NM (Negativo Mediano) y NL (Negativo Grande).

4) FAM (Fuzzy Asociative Memory) Estas FAM asocian la regla donde se relaciona a los conjuntos difusos B con los conjuntos difusos A.

5) TOMA DE DECISIONES. Este punto se refiere a los mecanismos de inferencia, basados en la definición de la implicación difusa. De acuerdo a los diferentes procedimientos de reglas de inferencia se puede tener los siguientes: De acuerdo al método de Producto Máximo considerando las variables de entrada X y Y.

5 a) Método Gráfico MAX-MIN de inferencia

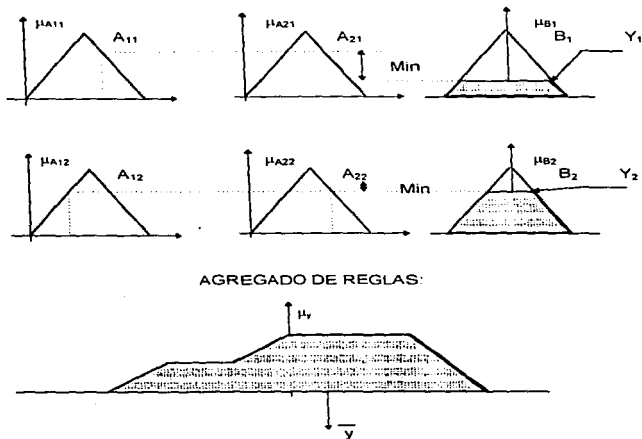


Fig.2.25 Agregado de reglas.

$$\mu_y(y) = \text{MAX}_k \{ \text{MIN} [\mu_{A_1} k(X_1), \mu_{A_2} k(X_2), \mu_B(y)] \} \quad y = \text{Defus. } [\mu_y(y)] \quad \text{ec. 2.37}$$

5.b) Método Gráfico de Producto Máximo

$$\mu_y(y) = \text{MAX}_k \{ [\mu_{A_1} k(X_1) \mu_{A_2} k(X_2) \mu_B k(y)] \dots \} \quad y = \text{Defus. } [\mu_y(y)] \quad \text{ec. 2.38}$$

Regla 1

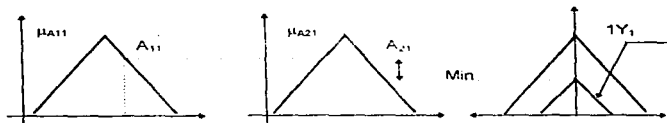


Fig. 2.26 Reglas para el método del producto máximo.

5.c) Método Gráfico MÁX-MÍN para entradas fusificadas

$$y = \text{Defus. } [\mu_y(y)]; \mu_y(y) = \text{MAX}_k \{ \text{MAX}_{x_1 \in A_1} \{ \text{MAX}_{x_2 \in A_2} \{ \text{MIN} [\mu_{A_1}(X_1), \mu_{A_2}(X_2), \mu_{A_1} k(X_1), \mu_{A_2} k(X_2), \mu_B k(y)] \} \} \} \} \} \quad \text{ec. 2.39}$$

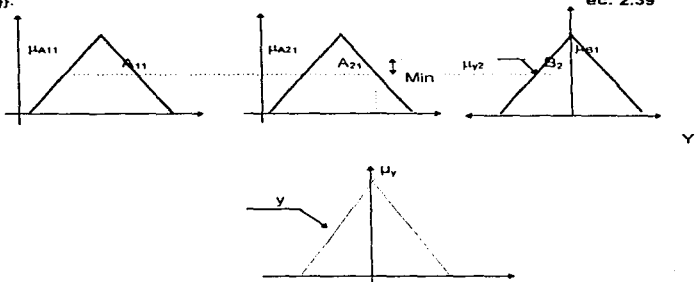


Fig. 2.27 Reglas para el método máx. mín. para entradas fusificadas

5.d) Método Gráfico de producto Máximo para Entradas Fusificadas.

$$\mu_V(y) = \text{MAX}_k \{ \{ \mu_{A1}(x_1) \mu_{A2}(x_2) \} \}$$

ec. 2.40

$$Y = \text{DEFUS. } [\mu_V(y)]$$

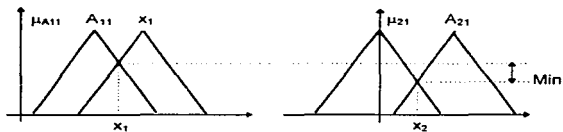


Fig. 2.28 Producto máximo para entradas fusificadas

CAPITULO 3

ESTRUCTURA ALGORITMICA DEL CONTROLADOR DIFUSO AUTOSINTONIZABLE.

3.1 INTRODUCCION

Después de haber visto en el capítulo anterior las ventajas que da el utilizar lógica difusa, en este procederemos a realizar la elección del algoritmo de control difuso que tenga las características apropiadas para la solución óptima de nuestro sistema. Además, daremos una explicación de como fue diseñado el algoritmo, mencionaremos sus características más importantes y justificaremos su uso con algunas pruebas que hicimos durante la etapa de diseño del sistema.

Además de los algoritmos basados en lógica difusa es importante mencionar que existen otros tipos de algoritmos de control, que son los que hasta la fecha se han venido usando con buenos resultados, aunque tienen algunas desventajas con respecto a los algoritmos difusos, que los hacen más difíciles de implementar en arquitecturas como la que proponemos para nuestro sistema de la cual daremos una amplia explicación en el siguiente capítulo.

3.2 CARACTERISTICAS Y CLASIFICACIONES DE LOS ALGORITMOS DE CONTROL.

Existen varios tipos de algoritmos de control con características y ventajas específicas para cada uno de ellos, que los hacen muy útiles para cierto tipo de aplicaciones

Haciendo una primera clasificación podemos decir que los algoritmos de control pueden ser de lazo abierto o de lazo cerrado, lo cual dependerá de la estructura de control que se tenga, así como del proceso a controlar. Una segunda clasificación se puede hacer dependiendo del modelo matemático del controlador, ya sea si es un modelo continuo o un modelo discreto. Otra clasificación se basa en la estructura de control utilizada, de donde surgen: controladores proporcionales, integrales, derivativos, adaptables, robustos, autosintonizables, difusos, etc. Cada uno puede tener más de una característica y ser utilizado para resolver determinadas tareas, de acuerdo con su diseño.

3.2.1 CARACTERISTICAS DE LOS ALGORITMOS DE CONTROL.

Desde el punto de vista matemático cualquier algoritmo de control podrá representarse por expresiones matemáticas, ecuaciones diferenciales para el caso de sistema continuo y ecuaciones en diferencias para el caso de un sistema discreto. Excepto el controlador difuso que no es representado por un modelo matemático convencional si no por reglas de inferencia, las cuales son representadas por un modelo discreto para realizar el análisis del comportamiento del sistema. De lo anterior podemos mencionar que un controlador convencional (no difuso), tiene la ventaja de contar con una precisión adecuada que se deriva del análisis al modelo matemático del proceso a controlar y de la estructura matemática del controlador, y la desventaja es que se necesita conocer el modelo matemático del proceso a controlar. En cambio un controlador difuso tiene la ventaja de no necesitar el modelo matemático que describa el comportamiento del proceso, aunque tiene la desventaja de que en ocasiones necesita un número grande de reglas de inferencia que simulan el comportamiento del proceso, teniendo en cuenta que entre más reglas existan se necesitará más tiempo de cálculo y más espacio de memoria [8].

A continuación la figura 3.1 muestra un modelo discreto que servirá de base para el diseño del controlador difuso utilizado en esta tesis, el diseño del controlador lo haremos más adelante en este capítulo [18].

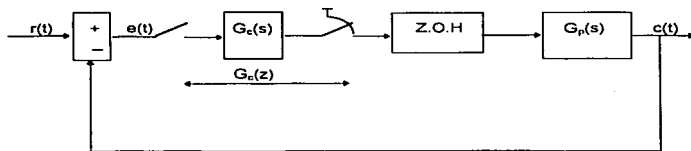


Fig. 3.1 Controlador digital con controlador de lazo en cascada

En la figura 3.1 se pueden ver tanto la salida como la entrada, que son las funciones continuas $r(t)$ y $c(t)$ respectivamente, estando la salida retroalimentada a la entrada, lo cual genera una señal de error $e(t)$, que es la diferencia entre la señal de entrada y la salida retroalimentada. El controlador digital esta representado por la función $G(z)$ conectado directamente al proceso $G(S)$ a controlar, el cual es una función continua.

Este esquema con las modificaciones adecuadas tiene la estructura de control básica utilizada en la tesis.

3.3 DESCRIPCION GENERAL DE LA ESTRUCTURA DEL ALGORITMO DE CONTROL UTILIZADO.

Debido a las características del proyecto tuvimos la necesidad de adaptar nuestro algoritmo de control, para cubrir de manera óptima las necesidades del sistema. Dicho sistema es un controlador de presión y temperatura, implementado en una arquitectura de procesamiento tipo *PIPELINE*, formada por tres etapas principales, entre las que se distribuyen las tareas que realiza nuestro algoritmo. En base a lo anterior y a las características específicas de cada etapa de nuestro sistema, pensamos que la manera más eficiente de asignar tareas es la que proponemos, ya que las tres etapas están trabajando de forma sincronizada y siempre se encuentran realizando alguna tarea, con lo que se gana en tiempo de procesamiento, además se optimizó el uso de memoria al máximo para hacer lo mas eficiente posible al sistema.

Es importante recordar que como se vió en el capítulo anterior, un algoritmo de control basado en lógica difusa consta de 3 etapas principales que son la *FUSIFICACION*, *EVALUACION DE REGLAS* y la *DEFUSIFICACION* y una cuarta no menos importante que es la *AUTOSINTONIZACION*, las cuales fueron asignadas en las tres etapas de nuestra arquitectura, bajo los criterios de hacer mínimo el tiempo de procesamiento y optimizar el uso del espacio en memoria para así poder catalogar a nuestro sistema como un sistema eficiente.

La figura 3.2, muestra a manera de bloque, las principales etapas del algoritmo de control y como es que quedaron distribuidas en la arquitectura. En esta podemos ver muy claramente como esta conformado nuestro sistema. Por la etapa 1 se reciben los datos de presión y temperatura, a los cuales se les realiza el proceso de fusificación, también en esta etapa se determina el mínimo grado de pertenencia. Se pasan a la etapa 2, en las que se realiza el proceso de evaluación de reglas y defusificación respectivamente, para entregar los resultados a la etapa 3, en la que se realiza la autosintonización y se atiende a la PC enviándole los datos que esta solicite (mediante un proceso de interrupción). Además genera las salidas del sistema, que son el manejo de 4 actuadores (2 de presión y 2 de temperatura).

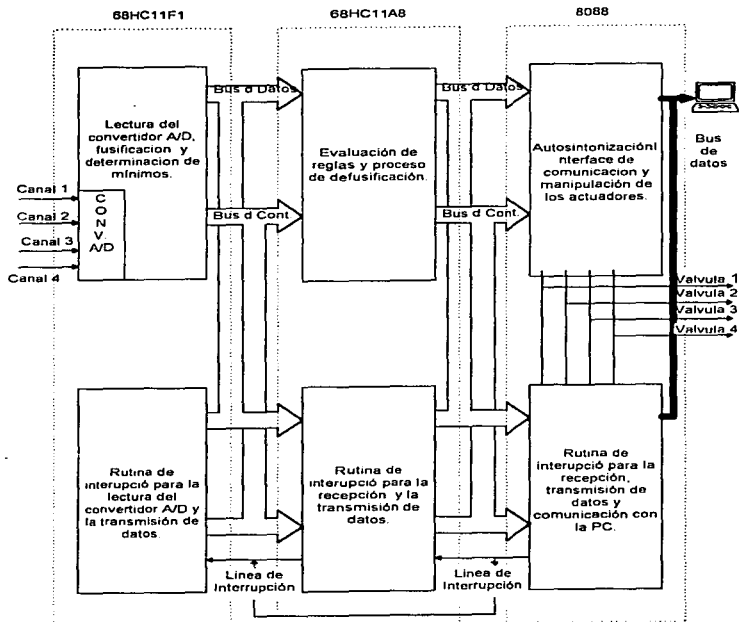
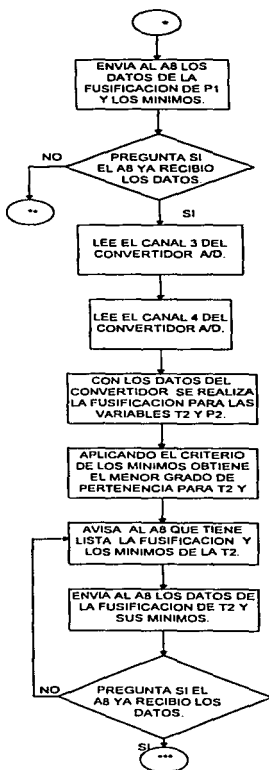
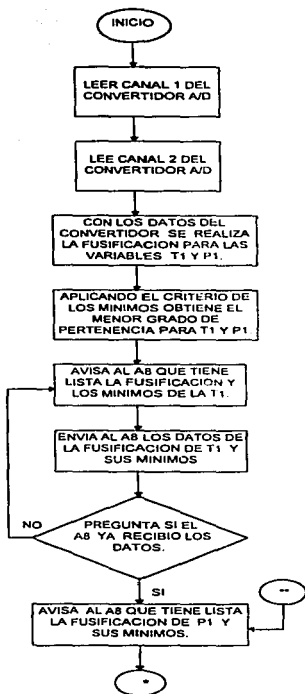


Fig. 3.2 Diagrama a bloques de la arquitectura del sistema.



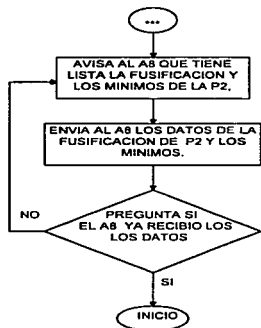
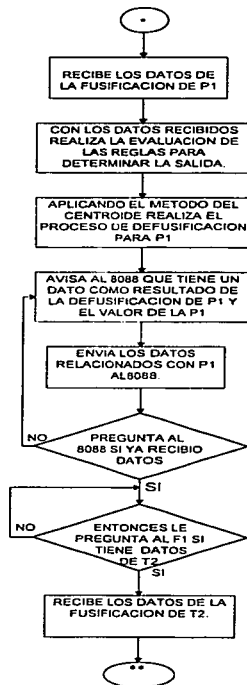
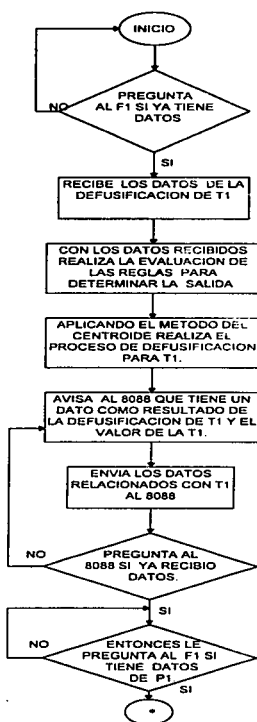


Fig. 3.2 b Diagrama de flujo del algoritmo cargado en el 68HC11F1.



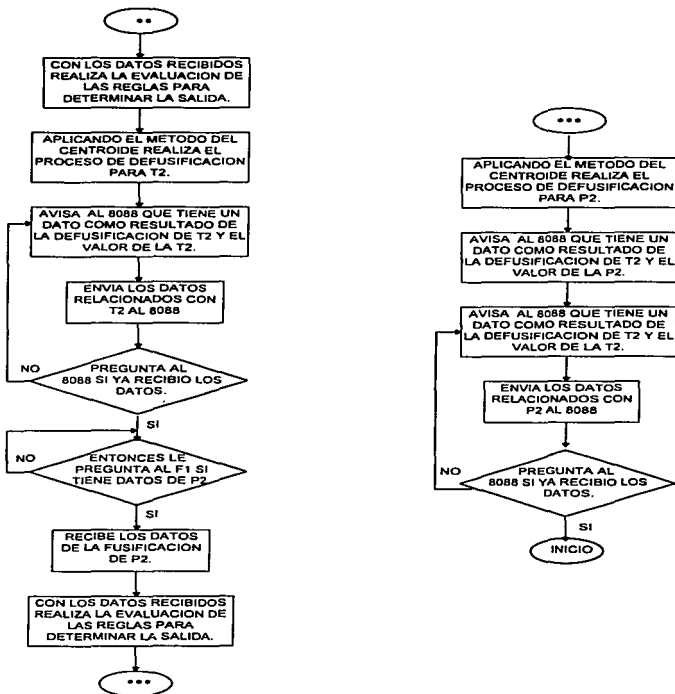


Fig. 3.2 c Diagrama de flujo del algoritmo cargado en el 68HC11A8.

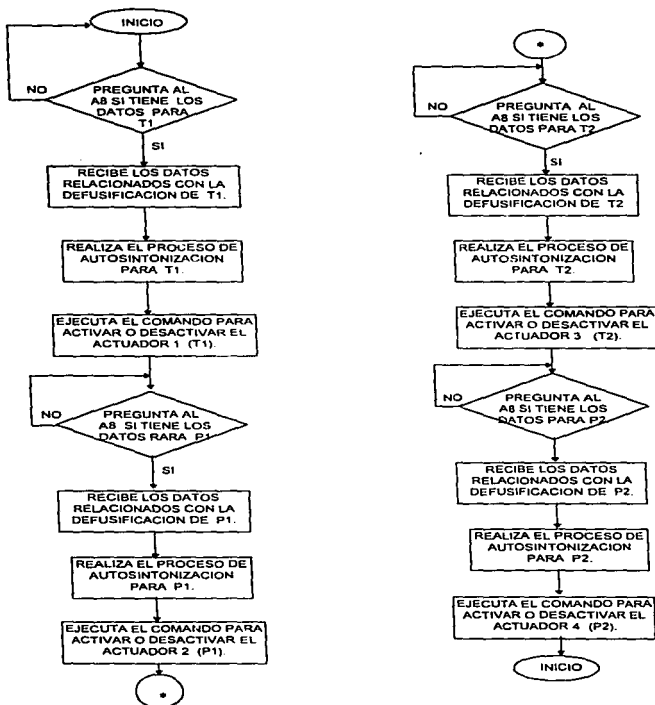


Fig. 3.2 d Diagrama de flujo del 8088 de Intel.

3.3.1 CARACTERISTICAS PRINCIPALES DE CADA ETAPA DEL PROCESO.

En esta parte podemos ver ya en forma conjunta las etapas del algoritmo y las etapas de la arquitectura como etapas del sistema de control o etapas del proceso, las cuales tienen las siguientes características:

ETAPA 1: Esta etapa es realizada por el primero de los procesadores y esta formada por la toma de valores de las variables, fusificación de los datos de entrada, asignación de las regiones a las que pertenece y obtención de los mínimos, que es una primera parte de la defusificación.

ETAPA 2: Se realiza en el segundo procesador y su función es tomar los datos del primer procesador para llevar a cabo la evaluación de las reglas de inferencia y acabar con la defusificación de los datos.

ETAPA 3: A esta etapa que se realiza en el tercer procesador, le corresponde la tarea de autosintonizar el dato defusificado que le envía la etapa 2, para generar con esto la salida para los actuadores del sistema, además está atendiendo a la PC para enviarle en formato serie los datos que esta le pide.

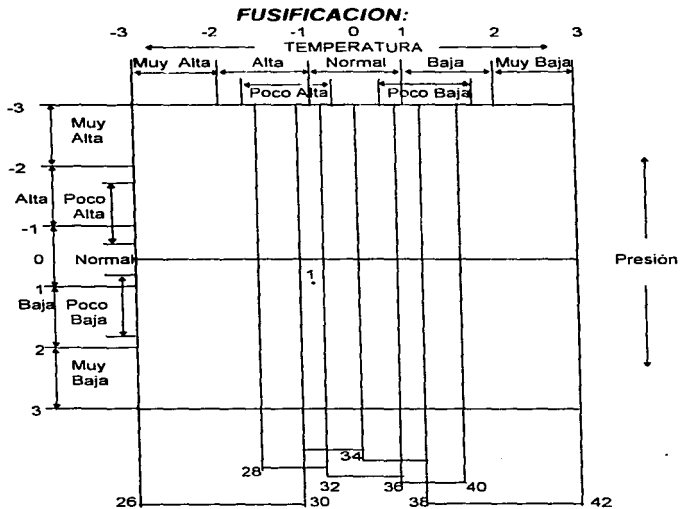
3.3.2 DESCRIPCION GENERAL DEL PROCESO.

El proceso está diseñado para el control y monitoreo de presión y temperatura, usando para esto cuatro sensores, 2 de temperatura y 2 de presión, los cuales se conectarán a la etapa 1, los valores de temperatura nos darán la oportunidad de manejar dos actuadores conectados a la etapa 3, los cuales tienen la responsabilidad de manejar el encendido o apagado de un calefactor y/o un ventilador respectivamente, por otra parte los sensores de presión también conectados en la etapa 1, nos dan la oportunidad de controlar dos actuadores también en la etapa 3, y que están conectados a un compresor y a una válvula, todo esto, para mantener la presión y la temperatura de una autoclave en un valor determinado. Además por medio de la etapa 3 se interactúa con el monitor central, el cual a través de comandos especiales solicita información del estado de las variables al controlador o toma por tiempo indefinido el control del proceso [23]

3.4 ALGORITMO DIFUSO UTILIZADO EN EL CONTROL DEL PROCESO.

Es importante mencionar que el método para la fusificación será el de triángulos y el método para la defusificación será el de centroides usando el criterio de los mínimos y que para la autosintonización utilizamos el criterio de estabilidad de Liapunov [17]. A manera de descripción mencionaremos en forma general como actúan las variables de entrada y como se comportan las salidas. Las entradas están agrupadas en pares, una temperatura y una presión y de acuerdo a los valores que tome un par de entradas (presión y temperatura) se tienen

solamente dos posibles valores para las salidas (actuadores), que son prendido o apagado. Estos algoritmos tendrán la siguiente estructura.



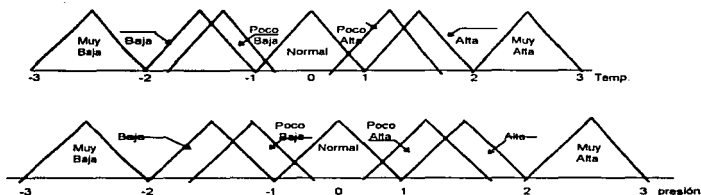


Fig. 3.3 Matriz de conjuntos difusos y método de fusificación para una de nuestras variables

3.4.1 PROCESO DE FUSIFICACION.

El proceso de fusificación consiste en trasladar las variables de entrada a conjuntos difusos asignando una variable lingüística a cada uno de ellos, realizándose este procedimiento para cada uno de los datos de entrada.

Para nuestro proceso las variables de entrada son 2 Temperaturas y 2 Presiones, las cuales se fusificaron quedando sus conjuntos difusos de la siguiente forma:

ENTRADAS.

TEMPERATURAS 1 Y 2	PRESIONES 1 Y 2
Muy baja=MB	Muy baja=MB
Baja=B	Baja=B
Poco baja=PB	Poco baja=PB
Normal=N	Normal=N
Poco alta=PA	Poco alta=PA
Alta=A	Alta=A
Muy alta=MA	Muy alta=MA

Tabla 3.1 Conjuntos difusos de entrada

SALIDAS.

TEMPERATURAS 1 Y 2.	PRESIONES 1 Y 2.
Muy fría= 1	Vacío a
Fría 2	Semi vacío b
Templado bajo= 3	Poco adec. (-)= c
Adecuado= 4	Adecuado= d
Templado alto= 5	Poco adec. (+)= e
Caliente= 6	Lleno= f
Muy caliente= 7	Peligro= g

Tabla 3.2 conjuntos difusos de salida

3.4.2 GENERACIÓN DE REGLAS DE INFERENCIA.

Las reglas de inferencia se generan en base a la forma en que se desee que se comporte la salida. Un aspecto muy importante para la generación de reglas es el relacionado con la experiencia que se tenga en el manejo de la variable de salida, por lo que para esta etapa del proceso es de suma importancia contar con el apoyo de un experto (operador del sistema), el cual basado en su experiencia pueda decirnos como es que se comportará la salida para cada una de las condiciones a la entrada. Hay un detalle importante en esto y es que el experto no necesita ser Ingeniero ni saber de algoritmos matemáticos para poder colaborar en la solución del problema [2].

Las reglas que estamos utilizando son del tipo if then teniendo como condiciones de entrada, una temperatura y una presión y como consecuencia a la salida la condición del estado del actuador de presión o de temperatura.

Ejemplo de algunas de las reglas generadas para nuestro sistema teniendo como salida la presión:

- Si la temperatura es MB y la presión es MB entonces la salida es 1.
- Si la temperatura es MB y la presión es B entonces la salida es 1.
- Si la temperatura es B y la presión es PBI entonces la salida es 2.
- Si la temperatura es B y la presión es N entonces la salida es 2.
- Si la temperatura es PB y la presión es PA entonces la salida es 3.

La figura 3.4, muestra la estructura de las reglas del proceso para un par de variables de entrada y como salida la presión. Se plantea una matriz cuadrada, de tal manera que en los renglones se encuentran los conjuntos difusos relacionados con la presión, en las columnas los relacionados con la temperatura y en las intersecciones se localizan los conjuntos difusos para la salida. Este ejemplo se realizó para obtener una salida de Presión.

		TEMPERATURA 1						
		MB	B	PB	N	PA	A	MA
P R E S I O N 1	MB	1	2	3	4	5	6	7
	B	1	2	3	4	5	6	7
	PB	1	2	3	4	5	6	7
	N	1	2	3	4	5	6	7
	PA	1	2	3	4	5	6	7
	A	1	2	3	4	5	6	7
	MA	1	2	3	4	5	6	7

Fig. 3.4 Matriz de Reglas de Inferencia para la Presión.1.

Como podemos apreciar en la figura 3.3, existen regiones que se traslapan, de tal manera que habrá valores que pertenezcan tanto a una como a otra - esto visto desde el punto de vista geométrico - Pero desde el punto de vista matemático tendremos como consecuencia un incremento en la exactitud ya que al momento de realizar la defusificación y obtener el centroide, contaremos con una gran cantidad de datos a procesar, lo que nos permitirá tener una mayor precisión en el valor de salida. Para el caso en el que deseamos tener como salida a la temperatura, sucede exactamente lo mismo que para el anterior, con la característica de que las regiones de salida no son los números si no las letras (a,b,c,d,e,f,g).

Con el valor de las dos variables de entrada de acuerdo a la matriz de reglas de inferencia tendremos un correspondiente valor de salida. Dichas salidas estarán divididas por regiones, las cuales a su vez estarán limitadas por valores de unidades de presión (Psi), dichos regiones se encuentran traslapadas, obteniendo con esto una mayor cobertura en la consecuencia de salida. Es importante hacer notar que entre más traslapes existan se tendrá una mayor exactitud en el valor de salida

3.4.3 DETERMINACIÓN DE LAS ALTURAS EN BASE A LA FUSIFICACIÓN DE LAS ENTRADAS.

El método que se siguió se basa en el criterio de mínimos visto en el capítulo 2, el cual nos dice que escogamos el mínimo valor de las alturas correspondientes al valor numérico de la variable de entrada. Es decir, a partir del valor que toma la variable de entrada trazamos una línea vertical para intersectar la o las regiones a las que pertenece obteniendo la o las alturas correspondientes a dicho valor numérico, por último, dichas alturas se compararán (las alturas correspondientes a una variable con las alturas correspondientes a la otra), para obtener la mínima.

Ejemplo:

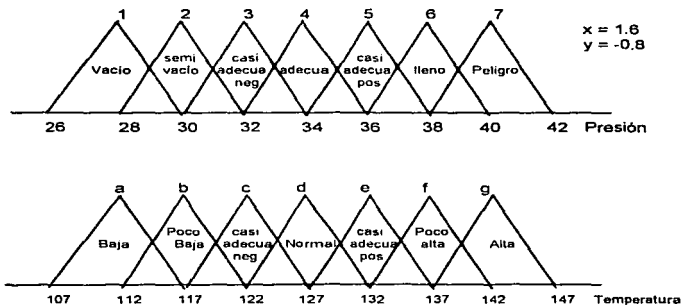


Fig. 3.5 Conjuntos difusos para nuestras variables de salida

DEFUSIFICACION:

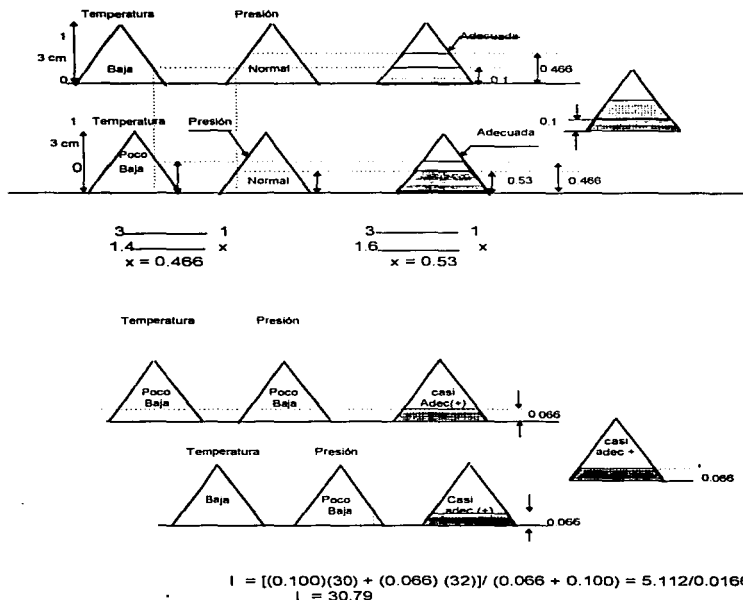


Fig. 3.6 Ejemplo de defusificación para una de nuestras variables en un punto determinado

3.4.4 DETERMINACIÓN DEL VALOR NUMÉRICO DE LA SALIDA EN BASE A LA DEFUSIFICACIÓN DE LAS ALTURAS DE LAS ENTRADAS

Una vez obtenidas las alturas mínimas se procede a determinar de acuerdo a las reglas de inferencia a que región de salida pertenecerá el dato, para que una vez determinada esta región, se le asigne la misma altura que fué determinada en el punto anterior. Las alturas asignadas a los triángulos de salida se compararán aquellas que pertenezcan a la misma región, para determinar las mínimas, ya que con estas se calculará el centroide que no será otra cosa que el valor numérico de salida del algoritmo.

3.4.5 ESTABILIDAD DEL CONTROLADOR DIFUSO.

Basicamente, la estabilidad de los controladores difusos estará determinada por la estabilidad de cada una de las reglas, de inferencia que intervienen en el algoritmo, de tal manera que si cada una de sus reglas es estable y a su vez en conjunto estas son estables, podemos decir entonces el sistema será estable [17].

3.4.6.- AUTOSINTONIZACIÓN DEL CONTROLADOR DIFUSO

La autosintonización del controlador estará determinada por la autosintonización de cada una de sus reglas, las cuales se plantearon en párrafos anteriores. Aquí analizaremos la autosintonización de una de las reglas de inferencia, dicha autosintonización es de primer orden, ya que solo se tomará la diferencial de la señal obtenida y la señal estimada. Esto funciona de la siguiente forma:

La variable de salida **X** después del proceso de defusificación tendrá un valor el cual será nuestra señal de salida, esta se comparará con dos valores que serán nuestra señal estimada, dichos valores son los centros de dos regiones, uno a la derecha y otro a la izquierda del valor de salida, de tal forma que un valor será mayor y el otro menor, con esto se determinarán dos diferencias como resultado de las comparaciones entre el valor de salida y los valores mencionados, la diferencia menor en valor absoluto es la que se tomará en cuenta ya que nuestra señal estará más cerca del correspondiente valor y tomará este como la verdadera salida, quedando de esta forma autosintonizado. Esto se hace con la finalidad de dar el valor exacto del centro de la región correspondiente ya que el sistema sólo tiene contemplados los centros de las regiones para el proceso de defusificación.

CAPITULO 4

DISEÑO DE LA ARQUITECTURA DEL SISTEMA.

4.1 INTRODUCCION

En relación a la arquitectura, en este capítulo nos enfocaremos a dar la explicación del porque de su diseño y de los elementos que la componen, así como sus ventajas y desventajas. Para cumplir con el objetivo del capítulo empezaremos por hacer un análisis a manera de referencia de lo que es el procesamiento paralelo, los tipos que existen, sus clasificaciones, sus ventajas y desventajas, enfocando nuestra atención en el procesamiento tipo PIPELINE. Después daremos la explicación de como esta diseñada nuestra arquitectura y explicaremos en que consiste cada una de las etapas que la integran, para que por último expliquemos los elementos que las componen.

Además hablaremos sobre microprocesadores y microcontroladores, diremos porque y como fueron seleccionados los que estamos usando, daremos las ventajas y desventajas que tiene cada uno, para que al final podamos tener una explicación del proyecto en conjunto, o sea del algoritmo de control y la arquitectura.

4.2 PROCESAMIENTO PARALELO.

4.2.1 PARALELISMO

El procesamiento paralelo es una manera eficaz de procesar información, que favorece la explotación de sucesos paralelos o traslapados de computación, además permite la ejecución de muchos programas, también de manera paralela, es un medio para mejorar el rendimiento del sistema mediante la realización de actividades simultaneas. Estos sucesos simultaneos se dan en una computadora en varios niveles de procesamiento [20].

- 1) Nivel de programación o trabajo.
- 2) Nivel de procesamientos o tareas.
- 3) Nivel de interinstrucciones.
- 4) Nivel de intrainstrucciones.

El nivel más alto de procesamiento paralelo se aplica a trabajos y programas múltiples, a través de la multiprogramación en el tiempo compartido y el multiprocesamiento. Este nivel

exige el desarrollo de algoritmos procesables en paralelo. La implementación de algoritmos paralelos depende de la asignación eficaz de recursos limitados de software y hardware.

El segundo nivel de procesamiento paralelo se aplica a procesamientos o tareas dentro del mismo programa. Esto supone la descomposición de un programa en múltiples tareas.

El tercer nivel trata de explotar la simultaneidad entre múltiples instrucciones. Con frecuencia se realiza un análisis de dependencia de datos para revelar paralelismo entre instrucciones

El balance entre las técnicas Hardware y Software para resolver un problema es siempre un asunto muy controvertido.

El cuarto nivel se implementa con frecuencia directamente por medios de Hardware. La participación del Hardware se va incrementando desde los niveles altos hasta los bajos.

4.2.2 CARACTERÍSTICAS DEL PROCESAMIENTO PARALELO.

La mayoría de los fabricantes de computadoras iniciaron el desarrollo de sistemas con un sólo microprocesador central denominados "SISTEMAS MONOPROCESADORES", los cuales tienen un límite refiriéndose al rendimiento. La potencia de cálculo de un monoprocesador, puede incrementarse si se permite el uso de elementos de procesamiento paralelo bajo el mando de un controlador [20].

También se puede ampliar la estructura de la computadora para incluir procesadores múltiples con espacio de memoria y periféricos compartidos bajo el control de un sistema operativo integrado. A este sistema de computadoras se les denomina "SISTEMA MULTIPROCESADOR" [20].

Por lo que se refiere a el procesamiento paralelo, la tendencia general de las arquitecturas se ha desplazado desde los sistemas monoprocesadores convencionales a los sistemas multiprocesadores, hasta una colección de elementos de proceso controlados por un monoprocesador. En todos los casos con un alto grado de segmentación encauzada en los distintos niveles del sistema.

4.3 CLASIFICACIONES DEL PROCESAMIENTO PARALELO.

4.3.1 POR SU ESTRUCTURA

Dependiendo de su estructura las computadoras que realizan procesamiento paralelo se clasifican como [20] :

a) **MONOPROCESADOR.** Una computadora monoprocador típica consta de tres componentes esenciales: la memoria principal, la unidad central de proceso (CPU), y el subsistema de entrada-salida.

b) **SEGMENTACION ENCAUZADA.** En un procesamiento de este tipo, la ejecución de una instrucción implica cuatro pasos principales: búsqueda de la instrucción (BI) desde la memoria principal; decodificación de la instrucción (DI), para identificar la operación a efectuar; búsqueda del operando (BO), si es preciso; y ejecución (EJ) de la operación aritmético-lógica decodificada. La fig. 4.1.a representa un procesamiento del tipo encauzado.

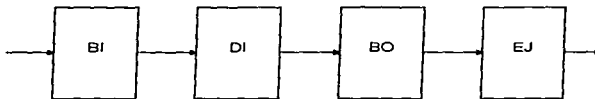


Fig. 4.1a Procesador encauzado

Las instrucciones sucesivas se ejecutan en modo traslapado como lo muestra la siguiente figura.

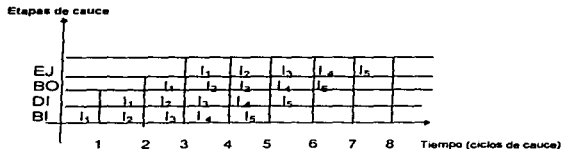


Fig. 4.1b Etapas de cauce de un procesador de segmentación encauzada

c) **MATRICIAL.** Un procesador matricial es una computadora paralelo síncrona con múltiples unidades aritmético-lógicas, llamadas elementos de proceso (EP), que pueden operar en paralelo. Por replicación de las ALU se puede conseguir paralelismo espacial. Los (EP) están sincronizados para efectuar todos la misma función al mismo tiempo. Entre los (EP) debe establecerse un mecanismo apropiado de encaminamiento de datos.

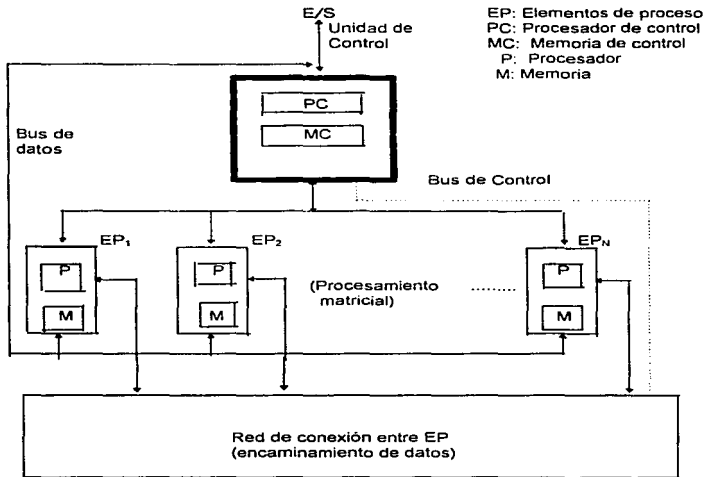


Fig. 4.2 Procesador matricial

Las instrucciones escalares y de control se ejecutan directamente en la unidad de control. Cada (EP) consta de una UAL (unidad aritmética lógica), con registros y memoria local, y se interconectan a través de una red de encaminamiento de datos

d) **SISTEMAS MULTIPROCESADORES.** La investigación y el desarrollo de los sistemas multiprocesadores se han enfocado a mejorar la productividad, confiabilidad, flexibilidad y disponibilidad de los sistemas [20].

Un sistema de este tipo contiene dos o más procesadores. Todos comparten acceso a grupos comunes de módulos de memoria, canales de entrada / salida y/o dispositivos periféricos. Lo más importante, es que el sistema entero debe estar controlado por un único sistema operativo integrado que facilite las interacciones entre los procesadores y sus programas a diferentes niveles.

La organización del sistema Hardware esta determinada principalmente por la estructura de interconexión entre las memorias y los procesadores (y entre la memoria y los dispositivos de E/S, si es necesario [20]).

BUS COMUN DE TIEMPO COMPARTIDO. El sistema de interconexión mas simple para varios procesadores es el que tiene un camino de comunicación común al que se conectan todas las unidades funcionales como se muestra en la siguiente figura.

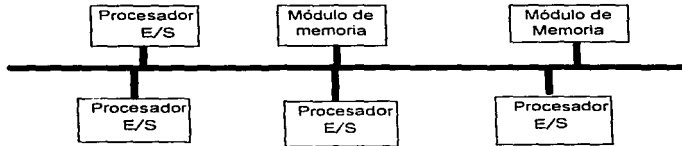


Fig. 4.3 Sistema multiprocesador de bus común de tiempo compartido.

El camino común se denomina frecuentemente bus común o de tiempo compartido. Esta organización es la menos compleja y la mas fácil de configurar. Una red de interconexión de este tipo suele ser con frecuencia una unidad totalmente pasiva, ya que no dispone de componentes activos como pueden ser los conmutadores. Las operaciones de transferencia son controladas completamente por las interfaces con el bus de las unidades de envío y recepción. Ya que el bus es un recurso compartido, se debe disponer de un mecanismo para resolver los conflictos que en él se presenten [20].

4.3.2 POR SU ALGORITMO.

Por su algoritmo el procesamiento en paralelo puede dividirse de la siguiente forma [20] :

- a) Clasificación de Flynn.
- b) Clasificación de Feng.
- c) Clasificación de Handler.

4.3.2.1 CLASIFICACION DE FLYNN (1966):

Esta clasificación se basa en la multiplicidad de los flujos de instrucciones y datos en un sistema de cómputo [20].

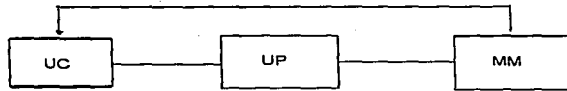
4.3.2.1.1 MULTIPLICIDAD DE FLUJOS DE INSTRUCCIONES Y DATOS.

Un flujo de instrucciones es una secuencia de instrucciones ejecutadas por la máquina; un flujo de datos es una secuencia de datos que incluye los datos de entrada y los resultados parciales o totales solicitados o producidos por el flujo de instrucciones. En general, las computadoras pueden clasificarse en cuatro categorías de acuerdo con la multiplicidad de los flujos de instrucciones y datos.

A continuación se listan las cuatro clasificaciones propuestas por Flynn:

- Simple flujo de instrucciones - Simple flujo de datos (SISD)
- Simple flujo de instrucciones - Múltiples flujos de datos (SIMD)
- Múltiples flujos de instrucciones - Simple flujo de datos (MISD)
- Múltiples flujos de instrucciones - Múltiples flujos de datos (MIMD)

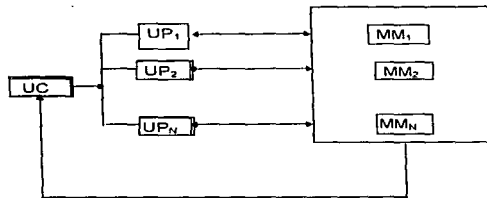
ORGANIZACION SISD. La organización mostrada en la figura siguiente representa la mayoría de las computadoras serie disponibles hoy en día. Las instrucciones se ejecutan secuencialmente pero pueden estar traslapadas en las etapas de ejecución. Una computadora SISD puede tener más de una unidad funcional, todas están bajo supervisión de una sola unidad de control.



UC: UNIDAD DE CONTROL
 UP: UNIDAD PROCESADORA
 MM: MÓDULO DE MEMORIA

Fig. 4.4 Computadora SISD

ORGANIZACION SIMD. Esta clase corresponde a los procesadores matriciales. Como se ilustra a continuación existen varios elementos de proceso (EP) supervisados por la misma unidad de control, que operan sobre diferentes conjuntos de datos procedentes de flujos distintos. El subsistema de memoria compartida puede contener múltiples módulos. Podemos dividir las máquinas SIMD a su vez en sección de palabra y sección de bit.



MC: memoria compartida
 FI: flujo de instrucciones
 FD: flujo de datos

Fig. 4.5 Computadora SIMD

ORGANIZACION MISD. Existen n unidades procesadoras; cada una con distintas instrucciones que operan sobre el mismo flujo de datos. Los resultados (LA SALIDA) de un procesador pasan a ser la entrada (los operandos) del siguiente procesador en el macrocauce.

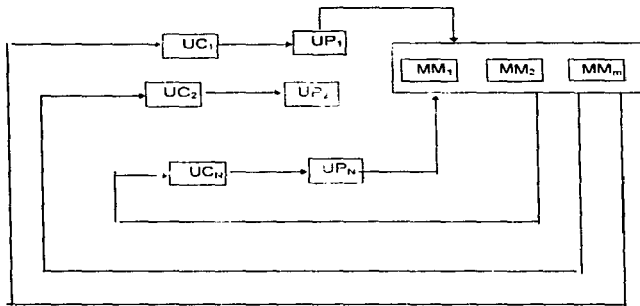


Fig. 4.6 Computadora MISD

ORGANIZACION MIMD. La mayoría de los sistemas multiprocesadores y de los sistemas con varias computadoras se incluyen en esta categoría. Una computadora MIMD intrínsecamente implica interacción entre los n procesadores porque todos los flujos de memoria se derivan del mismo espacio de datos compartido por todos los procesadores.

Si los n flujos de datos provinieran de subespacios disjuntos dentro de las memorias compartidas, entonces tendríamos la llamada operación SISD múltiple (MSISD), que no es otra cosa que un conjunto de n sistemas monoprocesadores SISD independientes. Una computadora MIMD está fuertemente acoplada si el grado de interacciones entre los procesadores es elevado, en caso contrario los consideramos débilmente acoplados.

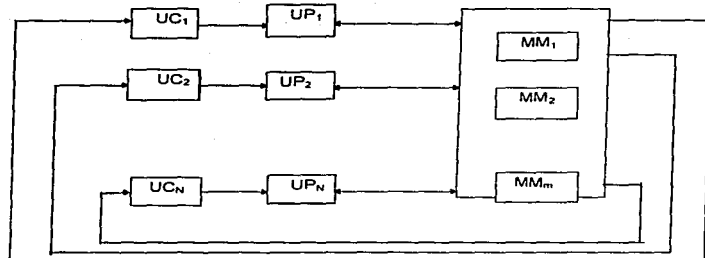


Fig. 4.7 Computadora MIMD

4.3.3. CLASIFICACION FENG

4.3.3.1 PROCESAMIENTO SERIE FRENTE A PROCESAMIENTO PARALELO.

Tse Yun Feng a sugerido el uso del grado de paralelismo como criterio de clasificación de las distintas arquitecturas de computadoras [20].

El número máximo de bits que pueden ser procesados en una unidad de tiempo por una sistema de cómputo se denomina MAXIMO GRADO DE PARALELISMO y se denota con la letra P [20].

Sea P_i el número de bits que pueden ser procesados en el i -ésimo ciclo del procesador. Consideremos T ciclos de procesador indicados por $i=1,2,\dots,T$. El grado medio de paralelismo P_m está definido por [20] :

$$P_m = \left(\sum_{i=1}^T P_i \right) / T \quad \text{ec. 4.1}$$

En general, $P_i \leq P_0$ por lo tanto, definimos la tasa de utilización " μ " de un sistema de cómputo durante T ciclos como:

$$\mu = P_m/P = \left(\sum_{i=1}^T P_i \right) / T \cdot P \quad \text{ec. 4.2}$$

Si el poder de cálculo del procesador está totalmente utilizada (o el paralelismo está totalmente aprovechado), tenemos entonces que $P_i = P$ para todo i y $\mu = 1$ con utilización del 100%.

La clasificación de las computadoras siguen el máximo grado de paralelismo que pueden alcanzar, se muestra en la siguiente figura. Sobre el eje horizontal se indica la longitud n de la palabra. El eje vertical corresponde a la longitud m de sección de bits. Ambas medidas de longitud vienen dadas en términos del número de bits contenidos en una palabra o en una sección de bits. Una sección de bits es una cadena de bits, una por cada una de las palabras sobre las que opera en paralelo [20].

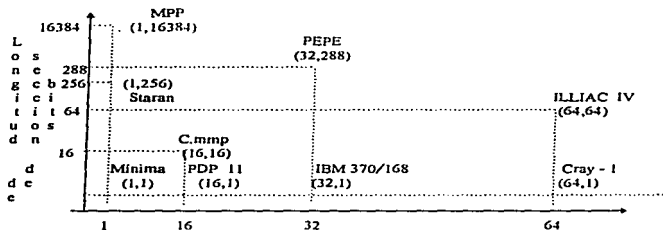


Fig. 4.8 Longitud de palabra

El máximo grado de paralelismo $P(C)$, de un sistema de cómputo C , está representado por el producto de la longitud n de palabra y la longitud m de sección de bit; es decir:

$$P(C) = n \cdot m.$$

ec. 4.3

Existen cuatro tipos de métodos de procesamiento que puede deducirse de este diagrama.

- a) Palabra serie y bit serie (PSBS) ($n = m = 1$)
- b) Palabra paralelo y bit serie (PPBS) ($n = 1$ y $m > 1$)
- c) Palabra serie y bit paralelo (PSBP) ($n > 1$ y $m = 1$)

El tipo **PSBS** ha sido llamado procesamiento bit-serie porque procesa un bit cada vez, lo cual resulta bastante lento. Se aplicó solamente en las computadoras de la primera generación [20].

El tipo **PPBS** ha sido llamado procesamiento por sección de bits porque procesa una sección de m bits cada vez [20].

PSBP es el tipo más frecuente en las computadoras actuales, ha sido llamado procesamiento por sección de palabra porque procesa una palabra de n bits cada vez [20].

El tipo **PPBP** se conoce como procesamiento totalmente paralelo y en él se procesa una matriz de $n \times m$ bits cada vez; es el modo más rápido de procesamiento [20].

4.3.4 CLASIFICACION DE HANDLER

4.3.4.1 PARALELISMO FRENTE A SEGMENTACION ENCAUZADA

Wolfgang Handler ha propuesto un esquema de clasificación para identificar el grado de paralelismo y el grado de encauzamiento contenido en las estructuras hardware de una computadora. Él considera el procesamiento paralelo encauzado en tres niveles de subsistemas [20].

- a) La unidad de control del procesador (CPU)
- b) La unidad aritmético lógica (ALU)
- c) El circuito a nivel de bit (CNB)

Cada CPU corresponde a un procesador. La ALU es equivalente al elemento de proceso (EP) que especificamos para los procesadores matriciales SIMD. El CNB corresponde a la circuitería lógica necesaria para efectuar operaciones de un bit en la ALU.

4.4 CARACTERÍSTICAS DE LA SEGMENTACIÓN ENCAUZADA.

La segmentación encauzada ofrece un modo económico de realizar paralelismo en las computadoras. El concepto de procesamiento encauzado dentro de una computadora es similar al de las líneas de montaje en una planta industrial. Para conseguir la segmentación encauzada, debe subdividirse la tarea (proceso) de entrada en una secuencia de subtareas, cada una de las cuales puede ser ejecutada por una etapa de hardware especializada que actúe en concurrencia con otras etapas de encauzamiento.

La segmentación de las tareas de entrada en una secuencia adecuada de subtareas resulta ser un factor crucial al determinar el rendimiento del encauzamiento [20].

En un encauzamiento con retardo uniforme todas las tareas tienen el mismo tiempo de proceso. La relación de procedencia de un conjunto de subtareas ($T_1, T_2, T_3, \dots, T_n$) de una determinada tarea T implica que una tarea T_i no puede comenzar hasta que una cierta tarea previa T_j ($i < j$) finalice, las interdependencias de todas las subtareas forman el grado de procedencia [20]. Con una relación de procedencia lineal, la tarea T_i no puede comenzar hasta que todas las subtareas previas acaben.

Un cauce lineal puede procesar una sucesión de subtareas con un grado de procedencia lineal. Un procesador de encauzamiento lineal básico se muestra en la siguiente figura. El cauce consiste en una cascada de etapas de proceso. Las etapas son circuitos combinacionales puros que efectúan operaciones aritméticas o lógicas sobre el flujo de datos que circulan a través del cauce. Las etapas se separan mediante registros de acoplamiento (latches) rápidos. Estos registros retienen los resultados intermedios entre las etapas. Los flujos de información entre etapas adyacentes están bajo el control de un reloj común aplicado a todos los registros de acoplamiento simultáneo [20].

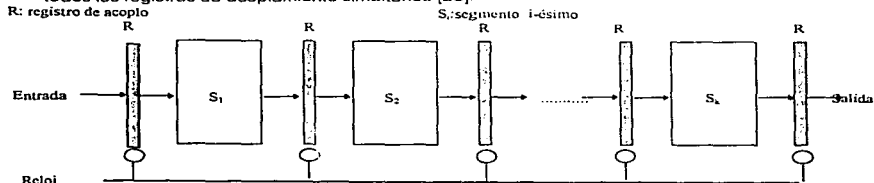


Fig. 4.9 Segmentación encauzada (cauce lineal).

La unidad central de procesamiento (CPU) puede partitionarse generalmente en tres secciones: La unidad de instrucciones, la cola de instrucciones y la unidad de ejecución. La unidad de instrucciones consta de etapas encauzadas para búsqueda de instrucción, decodificación de instrucción, cálculo de dirección de operando y búsqueda de operandos. La cola de instrucciones es un área de almacenamiento tipo PEPS (primero en entrar primero en salir) para instrucciones decodificadas y operandos ya disponibles. La unidad de ejecución puede contener múltiples unidades funcionales de segmentación encauzada para funciones aritmético lógicas. Las tres unidades están encauzadas como se muestra en la siguiente figura.

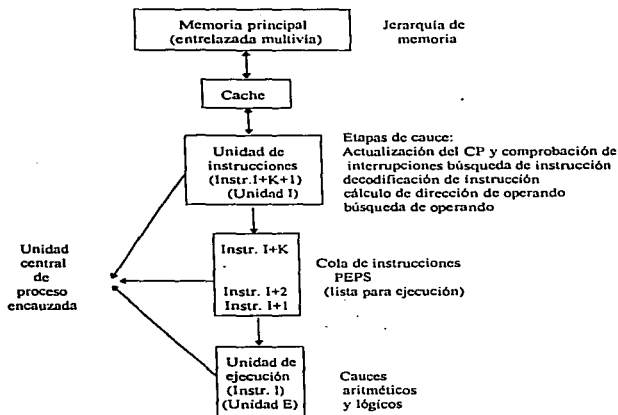


Fig. 4.10 Encausamiento de un procesador

4.4.1 CLASIFICACION DE PROCESADORES DE SEGMENTACION ENCAUZADA.

Según los niveles de procesamiento, Handler ha propuesto el siguiente esquema de clasificación para procesadores de segmentación encauzada [20]:

ENCAUZAMIENTO ARITMETICO. Las unidades aritmético lógicas en una computadora pueden segmentarse para realizar operaciones encauzadas en varios formatos de datos [20].

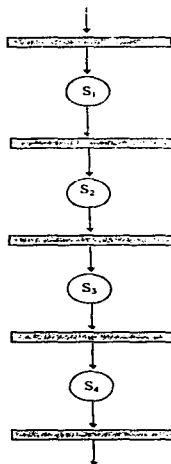


Fig. 4.11 Encauzamiento aritmético

ENCAUZAMIENTO DE INSTRUCCIONES. La ejecución de un flujo de instrucciones puede adoptar una estructura de segmentación encauzada que permita traslapar la ejecución de la instrucción actual con la búsqueda y decodificación de los operandos de las siguientes instrucciones [20]. Esta técnica se conoce también como anticipación de instrucciones.

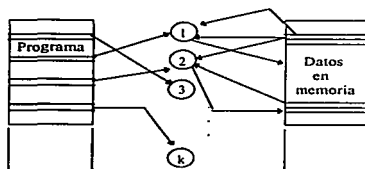


Fig 4.12 Encauzamiento de Instrucciones

ENCAUZAMIENTO DE PROCESADORES. Se refiere al procesamiento encauzado del mismo flujo de datos por parte de una cascada de procesadores, cada uno de los cuales procesa una tarea específica.

El flujo de datos pasa por el primer procesador y los resultados se almacenan en un bloque de memoria que es también accesible al segundo procesador. El segundo, pasa los resultados refinados al tercero y así sucesivamente.

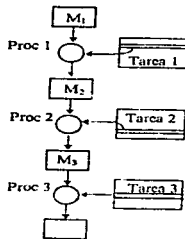


Fig. 4.13 Encauzamiento de procesadores

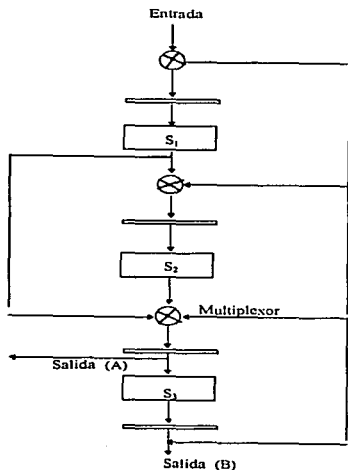
Dependiendo de las configuraciones del cauce y de las estrategias de control, Rama Moorthy y Li (1977), han propuesto los tres esquemas de clasificación siguientes [20]:

CAUCES UNIFUNCION O MULTIFUNCION. Una unidad encauzada con una función fija y dedicada, se denomina unifunción. Un cauce multifunción puede efectuar diferentes funciones, en diferentes momentos a la vez, mediante la interconexión de diferentes subconjuntos de etapas en el cauce.

CAUCES ESTATICOS Y DINAMICOS. Un cauce estático puede asumir una sola configuración funcional cada vez. Los cauces estáticos pueden ser uni o multifunción. El encauzamiento sólo es posible si se ejecutan continuamente instrucciones del mismo tipo.

Un procesador de encauzamiento dinámico permite la existencia simultánea de varias configuraciones funcionales. En este sentido un cauce dinámico debe ser multifuncional.

CAUCES ESCALARES O VECTORIALES. Un cauce escalar procesa una secuencia de operandos escalares bajo el control de un bucle DO son con frecuencia preextraídos y almacenados en la memoria temporal de instrucciones. Los cauces vectoriales están diseñados especialmente para manejar instrucciones vectoriales sobre operandos vectoriales. Las computadoras que disponen de instrucciones vectoriales suelen denominarse procesadores vectoriales. El diseño de un cauce vectorial es una extensión del diseño de un cauce escalar.



	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
S_1	A			A			A	
S_2		A						A
S_3			A		A	A		

	t_0	t_1	t_2	t_3	t_4	t_5	t_6
S_1	B				B		
S_2			B			B	
S_3		B		B			B

Fig. 4.14 Cauces escalares o vectoriales.

Las tablas de reserva son un diagrama bidimensional, tomado de las cartas de GANTT empleadas en investigación operativa, para mostrar como se utilizan las sucesivas etapas del cauce para la evaluación de una función específica en ciclos sucesivos de operación del cauce [20].

4.5 ¿PORQUE SEGMENTACION ENCAUZADA? (PIPE LINE)

Como se vio en los parrafos anteriores este tipo de procesamiento nos da la oportunidad de dividir el proceso en tareas, las cuales serán ejecutadas por una etapa de Hardware específica, que nos brinda una mejora muy grande en la ejecución del proceso.

El que podamos hacer una analogía de nuestro proceso con una línea de montaje, nos permite justificar el uso de este procesamiento. Como es bien sabido la subdivisión de trabajos en una línea de ensamble ha contribuido al éxito la productividad en masa en la industria, ya que el proceso central esta dividido en pequeños procesos haciendo más fácil la elaboración del producto final.

Esta filosofía de dividir en tareas un cierto proceso, es la base de nuestro sistema, ya que como hemos visto nuestro algoritmo esta formado por bloques que son fácilmente identificados, pudiendolos asignar a cada etapa del pipeline, logrando así que cada etapa del algoritmo pueda ejecutarse de forma más eficiente (Entendiendo como eficiencia la buena administración de la memoria y la velocidad del proceso).

Basados en lo anterior si superponemos un procesamiento encauzado como el de la figura 4.9, con la estructura de nuestro algoritmo dividido en bloques, obtendríamos un proceso como el de la siguiente figura:

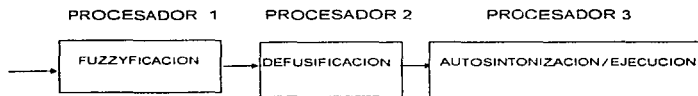


Fig. 4.15 Algoritmo difuso encauzado

Donde cada procesador representa a cada uno de los segmentos y su respectivo registro de acoplamiento de un encauzamiento lineal. Por lo tanto vemos que a cada bloque de segmentación encauzada le podemos asignar un bloque del algoritmo difuso, por eso se optó

por usar segmentación encauzada (pipe line). El siguiente diagrama espacio - tiempo nos ilustra como es que se encuentran traslapadas las operaciones de nuestro procesador:

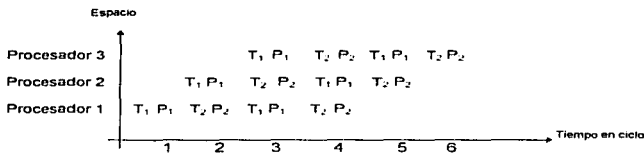


Fig. 4.16 Traslape de las instrucciones

Contrariamente, las implementaciones software, se incrementan desde los niveles bajos hasta los altos. Al disminuir los costos del hardware y aumentar los del software, los métodos hardware están reemplazando a los métodos software convencionales. La tendencia esta respaldada por la creciente demanda de tiempo real más rápido, compartición de recursos y entornos de computación tolerante a fallas.

Las características anteriores sugieren que el procedimiento paralelo es ciertamente un campo combinado de estudios. Requiere amplios conocimientos y experiencia en todos los aspectos de algoritmos, lenguajes, software, hardware, evaluación de rendimiento y alternativas de computación.

4.6 ARQUITECTURA DEL CONTROLADOR.

El controlador tiene dentro de su programación algoritmos de control difuso, que como ya se explico en los capitulos anteriores, se basan en reglas de inferencia las cuales nos permiten controlar le proceso sin tener que usar algún modelo matemático, como se había hecho mención, el controlador como tal está constituido por tres etapas con características especificas cada una (microcontrolador ó microprocesador), las cuales se encuentran realizando un procesamiento de tipo pipe line.

Principalmente la decisión de trabajar con arquitecturas de procesamiento en pipeline se basa en el factor "tiempo del proceso", ya que trabajando en esta forma se incrementará la velocidad del algoritmo y se reducirá el espacio de memoria utilizado.

Para nuestro caso en particular, la arquitectura utilizada es una arquitectura tipo pipeline, que está formada por dos HC11's y un microprocesador 8088. Donde el primer HC11 recibe las señales analógicas (variables de entrada), para realizar la fusificación y parte de la defusificación, asignando grados de pertenencia y obteniendo los mínimos, para transmitirlos al segundo HC11, y así este microcontrolador realice el resto de la defusificación, para que al terminar transmita los resultados al 8088, para que este realice la autotuning y mantenga comunicación con el monitor central [22], como lo muestra la siguiente figura [11], [19].

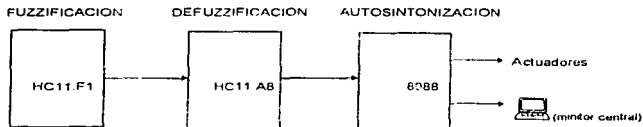


Fig. 4.17 Diagrama a bloques del controlador

4.7 MICROPROCESADORES Y MICROCONTROLADORES.

Desde la década de los 70's cuando aparecieron los primeros microprocesadores hasta nuestros días, su evolución ha sido demasiado rápida. Durante 1971, Intel produce el primer microprocesador, que era un dispositivo de 4 bits. Después diseña el 8008, el cual ya era un micro de 8 bits y tenía algunas ventajas sobre el anterior. Para 1974, aparecen varios fabricantes de microprocesadores y otros que ofrecían sistemas controlados por microprocesadores. Lo que trajo aunado la aparición de varios microprocesadores diferentes, los más comunes eran (algunos aún se utilizan) [11], [19].

INTEL:
8080, 8085

MOTOROLA:
6800

SIGNETICS:
6502

ZILOG:
Z80

Para 1978, aparecen los microprocesadores de 16 bits como el Intel 8086/8088, Motorola 68000 y Zilog Z8000, desde entonces los fabricantes han seguido desarrollando Microprocesadores con ventajas al que le precede y nuevas arquitecturas [11], [19].

Con este desarrollo muchas aplicaciones en control salieron beneficiadas, ya que por contar con un número reducido de componentes y una potencia de cálculo baja fácilmente podían ser sustituidos por un CPU. Así es como nacieron los microcontroladores, CPU's con memoria y

circuitos de interface internos. Uno de los mas recientes, que a nuestra opinión es de lo mejor, es el 68HC11, ya que es muy poderoso y además bastante fácil de usar en aplicaciones de CONTROL Y SU PRECIO ES BASTANTE BAJO [19].

Este microcontrolador apareció en el año de 1985, fue creado por Motorola y es compatible con su familia de microprocesadores 6800. Incluye algunas ventajas sobre los demás que lo hacen mejor, esas ventajas son:

- Utiliza tecnología HCMOS, lo que le permite ser muy rápido y pequeño en tamaño.
- Bajo consumo de potencia
- Alta tolerancia a fallas por señales de ruido, que es una característica muy importante para aplicaciones de control.

4.7.1 ESTRUCTURA INTERNA.

En esta parte por medio de la siguiente figura intentaremos explicar cuales son las características principales de un microprocesador y de un microcontrolador, en cuanto a su arquitectura interna se refiere. Para esto utilizaremos los diagramas a bloques de un microprocesador 8088 y de un microcontrolador 68HC11 [11], [19].

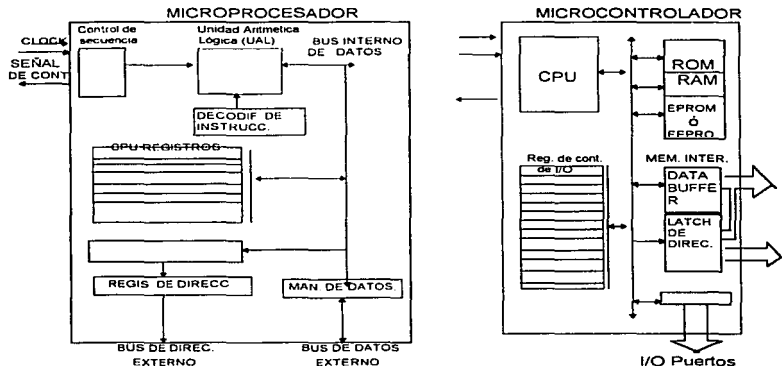


Fig. 4.18 Arquitectura Interna de Microprocesador Y Microcontrolador respectivamente.

Como podemos ver en la figura anterior, mientras en un microprocesador, el decodificador de instrucciones le dice a la ALU que hacer con el dato, el controlador de secuencia maneja la transferencia de instrucciones y datos a la largo del bus de datos interno. El registro de direcciones fija la dirección del bus de direcciones. El bus de direcciones externo selecciona una localidad especificada en memoria. El manejador de las, condiciones de los datos señala el dato que será enviado de o desde la memoria o puertos de entrada/salida [11], [24]

El microcontrolador cuenta con tres partes basicas: el CPU, la memoria y los registros, todas conectadas por un bus interno. El CPU controla la operación interna del microcontrolador. La memoria es donde los datos y códigos de programa son almacenados. Y los registros son utilizados para almacenar información especial. Existen tres tipos básicos de registros de I/O: Datos, Control y Estatus. Estos registros junto con el CPU y los puertos de I/O son utilizados para que el Microcontrolador se pueda comunicar con el exterior [12], [19].

4.7.2 TIPOS Y CARACTERISTICAS.

Actualmente existen varios fabricantes de microprocesadores y microcontroladores que han diseñado circuitos con características especiales que los hacen ser diferentes, además de que tienen ventajas y desventajas uno con respecto a otro. Algunos de los microprocesadores y microcontroladores más comunes aparecen en la tabla número 1 (pagina siguiente). [1], [19], [24], [25]:

	INTEL		MOTOROLA		MICRICHIP	T.I.	OTROS
RELOJ .	8088 15 MHz	8031 12 MHz	68000 8 MHz	68HC11 8 MHz	PICXX 20 MHz	DSP56K	68HC16 16 MHz
Mem. Interna.	---	128 Bytes de RAM	---	256 Bytes de RAM	192 Bytes 4 K	256 Bytes de RAM X y 256 de RAM Y 3.7 kbytes P/prog	1024 Bytes de RAM
Mem. Externa	1 Mega	64 K	16 Megs	64 K	8 K	65 K p/X y 65 K p/Y	1 Mega p/ prog y 1 Mega p/datos
PUERTOS.	PPI y USART	4 Paralel y 1 Serie Asinc	---	1 Serie asinc y 1 sinc 5 Paralelos	33 pines c/ direc indiv	1 serie asinc y 1 serie sinc y 4 paralel	1 serie asinc y 1 serie sinc y 6 paralel
Tam. de Palab.	8 Bits	8 Bits	16 Bits	8 Bits	8 Bits	24 Bits	16 Bits
INTERUPC.	256 2 Son por Hardware	5 con dos niveles de prioridad	7 niveles de prioridad	17 con prioridades fijas	12 Niveles	20 c/ 4 niveles de prioridad	20 c niveles de prioridad
Temporizadores	---	2 de 1 2 bits	---	1 de 16 bits	3 de	---	4 de 16 bits
Convertidor A/D	---	---	---	4 canales de 8 bits c/u	8 canales de 8 bits	---	8 canales de 16 bits c/u
Acumuladores	4	2	8 de Datos y 8 de Direc.	7	2	2 de 56 bits c/u	---

Tabla 4.1 Características de algunos microprocesadores y microcontroladores

4.7.2.1 CARACTERISTICAS

RELOJ: La frecuencia a la que opera el dispositivo es conocida como reloj, en todos los microprocesadores es suministrada por medio de un dispositivo externo (cristal de cuarzo, circuito RC o por un circuito oscilador). Un periodo del reloj es conocido como un ciclo de máquina, por lo que para determinar el tiempo en que es ejecutada una instrucción se debe saber cuantos ciclos consume.

MEMORIA: Con respecto a su implementación, existen dos tipos de memoria; externa e interna. Un microprocesador solo cuenta con memoria externa, la cual puede ser de lectura, de escritura o ambas y es manejada mediante el bus de direcciones.

En un microcontrolador existen los dos tipos de memoria. La interna que es donde se encuentran todos sus registros, además cuenta con regiones de RAM, ROM y en algunos EEPROM para utilizarlas con programas pequeños. Su memoria externa es de mayor capacidad y puede ser de lectura, escritura o ambas, es direccionada por medio del bus de direcciones del micro y se usa para aplicaciones que requieran de un espacio mayor en memoria [11], [19].

PUERTOS: Los puertos son los elementos por medio de los cuales un microprocesador puede tener contacto con el mundo exterior. Son de dos tipos paralelo y serie, toman ese nombre de la forma en que transmiten o reciben los datos [11], [19].

TAMAÑO DE PALABRA: Se denomina así al número de bit's que conforman un dato que pueda ser manejado por el microprocesador. Además es el espacio (a lo ancho) que tiene la memoria [11], [12].

VELOCIDAD DE PROCESAMIENTO: Este concepto esta completamente relacionado con el reloj y con el diseño del dispositivo, y nos dice que tan rápido es el microprocesador para ejecutar una instrucción y estar listo para atender la siguiente [12], [24].

CONJUNTO DE INSTRUCCIONES: Es el grupo de instrucciones que es capaz de realizar un microprocesador. Para cada uno existen dentro de este grupo, instrucciones especiales que los hacen diferentes [11], [12].

CAPACIDAD DE MEMORIA: Es el máximo espacio disponible en un microprocesador para almacenar información (programas y/o datos) [11], [12].

4.7.2.2 CARACTERISTICAS ELECTRICAS.

La mayoría de los microprocesadores y microcontroladores son alimentados por medio de una fuente de voltaje de 5 Volts de DC \pm 10%. Manejan una corriente de aproximadamente 320 à 500 mA, operan a temperatura ambiente, salvo algunos diseñados para uso específico. La tecnología utilizada en su fabricación permite contar con dispositivos pequeños en tamaño y de bajo consumo de potencia.

4.7.3 CRITERIOS DE SELECCION.

Para el diseño de cualquier equipo electrónico es importante tomar en cuenta varios aspectos sobresalientes de los dispositivos a utilizar para hacer una elección adecuada.

Tenemos que entre las características más importantes a considerar se encuentran las siguientes:

COMPATIBILIDAD: Que quiere decir que los circuitos a usar cuenten con la característica de poder ser interconectados con otros de diferente fabricante y tecnología de diseño.

CAPACIDAD DE OPERACION: En esta característica están incluidos varios aspectos importantes, como son; capacidad de memoria, velocidad de procesamiento, velocidad de respuesta Fan in y Fan out, la posibilidad de operar dentro de un amplio rango de temperatura y el número de instrucciones que puede ejecutar [11], [12]

COSTO: Es importante hacer un buen análisis de costos, para que la inversión a realizar sea mínima, sin que esto implique que la calidad en el sistema sea baja o nos imponga limitantes en el diseño.

SOPORTE TECNICO: Para que los dispositivos puedan ser elegidos es muy importante conocer el respaldo bibliográfico del que disponemos, así como del software que necesitamos para el diseño del sistema. Otro aspecto que tampoco debemos olvidar y que no es menos importante, es el que se refiere a la disponibilidad en el mercado de las herramientas de desarrollo para facilitar su uso.

4.7.4 DISPOSITIVOS (MICROPROCESADORES) SELECCIONADOS.

Una vez que se tomaron en cuenta todos los puntos anteriores decidimos utilizar en nuestro sistema dos microcontroladores HC11 y un 8088. Los HC11 son un F1 y un A8. Como habíamos mencionado en párrafos anteriores, la arquitectura que estamos usando es del tipo Pipeline. En donde debido a su capacidad de memoria, a que incluye convertidor analógico digital y a su número de puertos, el primero de los micros en nuestro sistema es el 68HC11F1, el segundo que es también un 68HC11, modelo A8, tiene casi las mismas características que el anterior, solo que este es más pequeño en su capacidad de memoria. El último es el Intel 8088, un microprocesador muy poderoso y con las características apropiadas para realizar las tareas que le asignamos en el desarrollo del algoritmo.

A nuestro modo de ver en este momento surgen preguntas muy importantes, que debemos responder para justificar el uso de los microprocesadores que estamos usando. Las preguntas que nos parece importante responder es ¿por que usar estos microprocesadores? ¿por que se asigno ese orden? ¿Cuales fueron los criterios que se tomaron en cuenta para asignar las tareas que desarrolla cada microprocesador?

Los microprocesadores y microcontroladores que estamos usando fueron elegidos por sus características de operación y por su capacidad de ejecutar instrucciones que son básicas en nuestro sistema. Las ventajas que encontramos en estos micros son las siguientes:

68HC11 F1 Y 68HC11A8.	INTEL 8088.
- Frecuencia de operación	- Capacidad de memoria
- Convertidor A/D de 4 canales	- Velocidad de operación
- Capacidad de memoria	- Bajo consumo de corriente
- Bajo consumo de corriente	- Arquitectura fácil de adaptar a una con microprocesador más poderoso de la misma familia
- Fácil de programar	- Registros que pueden ser de 8 o 16 bits c/u
- Pequeño en tamaño	- Segmento de código y de datos separados
- Alta velocidad de operación	- Manejo de datos en varios formatos
- Alta tolerancia a fallos por señales de ruido	- 6 Modos de direccionamiento
- Adecuado para aplicaciones de control	- Amplio soporte técnico por ser de la familia del 80486
- Set de instrucciones muy completo	- Bajo costo
- Amplio soporte técnico	- Fácil de adquirir
- Bajo costo	- Puertos de entrada/salida fácil de programar
- Fácil de adquirir	- Existe la posibilidad de actualizar su arquitectura a un 80486 de ser necesario
- Cuenta con registros de 8 y 16 bits	- Amplio vector de interrupciones
- Tiene dos registros acumuladores	- Bus de datos y direcciones multiplexado
- 6 modos de direccionamiento	

Tabla 4.2 Ventajas que tienen los microprocesadores que se eligieron para el controlador.

Con respecto a dispositivos de su mismo tipo sus desventajas son pocas, se pueden enlistar en la siguiente tabla:

68HC11F1 Y 68HC 11A8	INTEL 8088.
- Algunas versiones tienen su bus de datos y direcciones multiplexados, por lo que al trabajar en modo expandido pierde algunos puertos.	- Necesita de un dispositivo externo para proporcionar la velocidad de transmisión al USART
- Algunas versiones no se consiguen tan fácilmente.	- No cuenta con instrucciones bit direccionables
	- Existe el riesgo de ruido al ejecutar una interrupción por hardware
	- Pequeño si lo comparamos con micros más modernos.
	- En desuso por la necesidad de micros más poderosos y rápidos.

Tabla 4.3 Desventajas de los microprocesadores que se eligieron para el controlador.

Como podemos ver son varias las características que hacen ideales, para nuestro sistema, a los dispositivos que estamos usando.

La segunda pregunta que debemos contestar es por que usar los microprocesadores en el orden que elegimos, la respuesta es simple, porque que se adaptaron a las necesidades de nuestro algoritmo, el cual se integra por tres partes fundamentales.

El F1 por contar con un convertidor analógico digital y tener la capacidad de direccionar un mayor espacio en memoria, es el primero. En él se lleva a cabo la primera parte del algoritmo y gran parte de la segunda, por lo tanto, la programación ocupa un mayor espacio, que el F1 es capaz de direccionar.

El siguiente microcontrolador es el A8, que cuenta con todas las características que tiene el anterior excepto que no tiene la capacidad de memoria que tiene el F1, por lo tanto solo ejecuta el resto de la segunda parte del algoritmo. Además nos es de gran utilidad en esta posición ya que además de realizar la segunda parte de la defusificación que consiste en evaluar las reglas y aplicar el método del centroide, debido a las instrucciones con que cuenta es muy hábil para mantener una buena comunicación entre el y los otros dos microprocesadores.

El tercero y último de los microprocesadores en nuestra arquitectura es el 8088, al cual se le asigno esta posición por las ventajas que tenemos al comunicarlo con una PC (que por lo general cuentan con microprocesadores de características similares), ya que los dispositivos que utiliza como interfaces de comunicación son totalmente compatibles con los que tienen las PC's. Además pertenece a la familia del 80486, lo que hace adaptable su arquitectura a una de mayor poder, más rápida y que cumple con las exigencias de velocidad y eficiencia de nuestros días.

La parte del algoritmo de control que ejecuta es la última (Autosintonización) y además maneja la comunicación con la PC a través de rutinas de servicio de interrupción. Las cuales son tareas que realiza fácilmente y de forma muy eficaz.

La última pregunta que nos hicimos es la relacionada con los criterios que tomamos para la elección de los micros y la respuesta esta implícita en las dos anteriores ya que respondiendo las anteriores en forma que quedemos satisfechos los criterios de selección pueden quedar implícitos en las respuestas.

4.8 DIAGRAMA A BLOQUES DE LA ARQUITECTURA.

En esta parte analizaremos el diagrama a bloques de la figura 3.2 del capítulo anterior, que muestra como esta formada nuestra arquitectura, dando una explicación de cada una de las partes que la integran.

Como podemos ver, esta arquitectura de tipo Pipe Line, esta dividida en tres etapas. La primera es una arquitectura mínima del 68HC11F1 en modo expandido, la cual tiene grabada la parte del algoritmo que realiza los siguientes pasos:

- a) Toma los datos del convertidor analógico digital (canal1=T₁ , canal2=P₁ , canal3=T₂ , canal4=P₂)
- b) Con los datos de entrada inicia el proceso de Fusificación (por parejas T₁ y P₁ ó T₂ y P₂)
- c) Determina las regiones a las que pertenece el dato, tanto a Temperatura como a Presión
- d) Una vez terminada la fusificación y asignadas las regiones, realiza la primera parte de la Defusificación que consiste en determinar por medio del método de los mínimos el menor de los grados de pertenencia para cada una de las regiones a las que pertenece el dato, tanto en temperatura como a presión
- e) Transmite los datos obtenidos al siguiente procesador
- f) Al final de la transmisión vuelve a empezar, y toma la siguiente muestra

La segunda parte esta formada también por una arquitectura mínima del 68HC11A8 en modo expandido, la cual se encuentra realizando los siguientes pasos:

- a) Recibe los datos que envió el primer procesador.
- b) Realiza la evaluación de reglas
- c) Determina la o las regiones de salida que corresponden a la entrada.
- d) Aplicando el método del centroide termina el proceso de Defusificación.
- e) Transmite los datos obtenidos al siguiente Procesador
- f) Una vez que terminó con un dato va al inicio y repite el procedimiento para el siguiente dato.

La tercera y última etapa del sistema esta formada por la arquitectura del 8088 de Intel, la cual tiene grabada la parte final del algoritmo que consta de los siguientes pasos:

- a) Recibe los datos que le envió el A8.
- b) Con el resultado de la defusificación inicia el proceso de autosintonización, el cual consiste en obtener las diferencias entre el dato y los centros de las regiones más próximas (a la derecha y a la izquierda del dato).

- c) Al terminar la autosintonización, en base al resultado se realiza la manipulación de las válvulas (actuadores), las que pueden tener dos estados (1 ó 0), encendido o apagado respectivamente.
- d) Una vez realizado, el proceso para uno de los datos, se inicia nuevamente el proceso para el siguiente dato.

Además de los pasos anteriores, que son parte del programa principal (algoritmo de control), esta arquitectura mantiene comunicación con la PC (monitor central), para enviarse los datos que le solicite (valor de las variables, estado de los actuadores), además recibe los comandos que desactivan o activan al controlador. Todo esto se encuentra dentro de una rutina de servicio de interrupción que es activada desde la PC. Esta parte se compone de los siguientes pasos [22]:

- a) Pregunta por el código que le envían para saber que tiene que hacer:

- Enviar datos (T₁, P₁, T₂ y P₂)
- Activar o desactivar el controlador
- Manipular actuadores

Una vez atendida la rutina de interrupción el procesador se saldrá al principio del programa para reiniciar el proceso.

4.9 PROTOCOLOS

Con referencia a este punto podemos mencionar que el protocolo de comunicación es un protocolo simple, tiene un formato de 9600 bauds, un bit de inicio, paquete de datos de 8 bits, y un bit de paro, sin bit de paridad, este formato incluye los códigos que permitan identificar en que momento se debe enviar la información, así como de cual es la información requerida, ya que tanto el procesador central como el controlador tendrán los códigos correspondientes. El código utilizado no tiene ningún formato de encriptado pero sí posee la característica de poder identificar y verificar mediante algoritmos difusos de comparación la información recibida como lo hacen algunos protocolos comerciales. A continuación se describirá la estructura de los protocolos de comunicación. De tal forma que para el proceso los códigos son: 06, 08 y 0C son los utilizados para la petición de datos, la desactivación del controlador y la activación del mismo respectivamente. De esta forma, se transmite el código ya sea del procesador central o hacia el procesador central en el formato antes mencionado, el controlador al recibir el código correspondiente responde con el mismo número de código, seguido de los datos correspondientes. Con esto en una escala numérica y una gráfica poder monitorear el proceso y en un momento dado poder mandar una orden de control si es necesario, teniendo como base un algoritmo en el procesador central igual al residente en el controlador externo [22].

El formato de los protocolos será el siguiente:

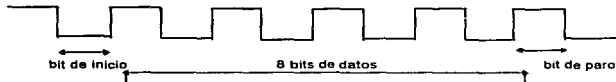


Fig. 4.19 Formato de los protocolos de comunicación

Como conclusión de este capítulo podemos mencionar que la combinación de una arquitectura **Pipeline** con los algoritmos de lógica difusa, nos dio grandes ventajas en la implementación de nuestro controlador, ya que hace que el software desarrollado sea más fácil, debido a que el algoritmo puede ser dividido en pequeñas tareas, lo que nos ofrece una mayor facilidad, además de ser más rápidos y utilizar un menor espacio de memoria.

CAPITULO 5

TRANSDUCTORES Y SENSORES.

5.1 INTRODUCCION.

Si deseamos que un proceso sea realmente controlado, es necesario que las variables del sistema sean monitoreadas. Esto es, las condiciones del sistema deben ser medidas constantemente y convertidas a una señal eléctrica representativa. Por ejemplo, para controlar la temperatura de una pieza de metal que está siendo maquinada, la temperatura del metal debe medirse constantemente. Con la información, la velocidad de los instrumentos de la máquina podrá ser ajustada para conservar la temperatura de la pieza maquinada dentro de un rango aceptable. Por lo tanto, partes integrantes de algunos sistemas de control de procesos son estos elementos o subsistemas que son capaces de sensar las condiciones del sistema. Estos elementos se pueden clasificar dentro de dos grupos:

5.2 TRANSDUCTORES DE ENTRADA Y SENSORES.

Técnicamente un transductor es un dispositivo capaz de transmitir energía de un sistema a otro. Como solo cambian las características eléctricas de un transductor, es necesario un circuito que soporte, detecte y amplifique estos cambios. Por consiguiente un transductor eléctrico de entrada solo no es un sensor completo.

Los sensores, por otro lado, están ensamblados de tal forma que comúnmente incluyen uno o más transductores, junto con todos los circuitos, necesarios para acondicionar la señal. La salida del sensor puede ser un contacto físico, un semiconductor, un voltaje o corriente alterna o algunos números binarios. Algunos sensores son tan sofisticados que no solo adquieren información acerca de las condiciones del sistema, si no que también forman y transmiten datos a computadoras.

Los transductores y los sensores se clasifican de acuerdo a la condición física que se está sensando con ellos. Los dispositivos más comunes caen dentro de las siguientes categorías [21]:

- TERMICOS
- OPTICOS
- MAGNETICOS
- ELECTROMAGNETICOS

5.3 TRANSDUCTORES Y SENSORES DE TEMPERATURA.

Son componentes que cambian sus características eléctricas de acuerdo con cambios en su temperatura. Pueden ser divididos dentro de tres grandes categorías [21] :

- 1.) Detectores de resistencia de temperatura (RTD)
- 2.) Termopares o Termocuplas.
- 3.) Termistores.

5.3.1 DETECTORES DE RESISTENCIA DE TEMPERATURA (RTDs).

Un detector resistivo de temperatura (RTD), está construido por bobinas de alambre enrolladas sobre formas de cerámica y vidrio o por el depósito de una película de metal en sustratos de cerámica. Tiene coeficiente de temperatura positivo, o sea que a medida que la temperatura aumenta, la resistencia del dispositivo también aumenta. Aunque los RTDs pueden ser construidos de cobre, níquel y algunas veces de tungsteno, el de platino es el más comúnmente usado, por ser fácil de purificar y ser químicamente estable.

Pueden ser usados generalmente desde aproximadamente 200°C hasta 700°C. Un RTD tiene una salida mucho más lineal que un termistor. De hecho, para pequeños rangos de temperatura, la respuesta puede ser considerada lineal. Como muchos han sido fabricados de Platino, sus características se han estandarizado.

Como la respuesta de un RTD es casi lineal, la siguiente ecuación puede ser usada para determinar la resistencia a temperaturas diferentes de 0°C.

$$R_t = R_0(1 + \alpha T) \quad \text{ec. 5.1}$$

donde

R_t = Resistencia a la temperatura deseada
 R_0 = Resistencia a cero grados Centígrados
 α = Coeficiente de temperatura del RTD
 T = Temperatura deseada.

Así, para un rango de temperatura de 0°C a 70°C, la resistencia del RTD puede variar de 100 a 369.5 Ω

Las ventajas y desventajas que se tienen al usar un RTD son las siguientes [21]:

VENTAJAS	DESVENTAJAS
1.) Linealidad sobre un gran rango de operación	1.) Baja sensibilidad
2.) Gran rango de temperatura de operación	2.) Costo mayor al de otros transductores de temp
3.) Alta temperatura de operación	3.) No requiere punta de soldado
4.) Mayor rango de cambio que otros transductores	4.) Puede ser afectado al golpearlo, transportarlo o tocar la resistencia
5.) Mayor estabilidad a altas temperaturas	5.) Requiere de tres o cuatro alambres de operación

Tabla 5.1 Ventajas y desventajas de los RTD's

La siguiente figura muestra la variación de la resistencia con respecto a la temperatura de algunos de los materiales mas comúnmente usados. La resistencia cambia con la temperatura que puede ser calculada por la ecuación:

$$R = R_0 (1 + \alpha_1 T + \alpha_2 T^2) \quad \text{ec. 5.2}$$

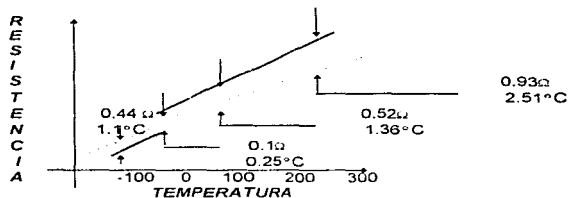


Fig. 5.1 Variación de la resistencia con respecto a la temperatura

5.3.2 EL TERMOPAR.

El Termopar es un transductor de temperatura que desarrolla una fuerza electromotriz que está en función de la diferencia de temperatura entre la unión caliente y la unión fría. Los Termopares hechos a base de metales nobles son comúnmente usados para medir temperaturas cerca de cero absoluto cerca de 3200°F, mientras unidades especiales están habilitadas para temperaturas de 5600°F.

Este dispositivo es el más común para medir temperatura en procesos industriales. Un termopar está formado por un par de conductores metálicos distintos unidos entre sí formando un circuito completo como se muestra en la figura 5.2.

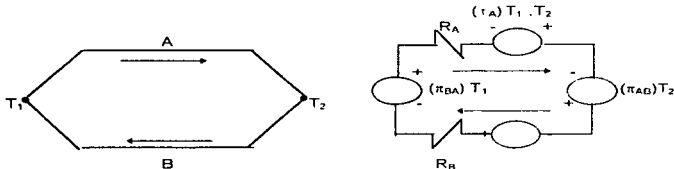


Fig. 5.2 Circuito formado por un termopar

Los conductores distintos tienen dos puntas de unión, una denominada la unión caliente, sometida a altas temperaturas y la otra la unión fría, sometida a una baja temperatura. Cuando se hace esto, se crea un pequeño voltaje neto en la malla; este voltaje es proporcional a la diferencia entre la temperatura de las dos uniones.

Lo que sucede es que se produce un pequeño voltaje en cada unión de los distintos metales, debido a un fenómeno denominado el efecto Seebeck [21]. Entre mayor sea la temperatura de la unión, mayor es el voltaje producido por dicha unión. Además, la relación entre el voltaje y la temperatura es aproximadamente lineal; es decir, un aumento en la temperatura produce un aumento en el voltaje. La constante de proporcionalidad entre el voltaje y la temperatura depende del par de metales que se utilice. Dado que una malla completa siempre tiene dos uniones, se producen dos voltajes. Estos voltajes se oponen entre sí en la malla. El voltaje neto disponible para manejar una corriente a través de la resistencia de la malla es la diferencia entre los dos voltajes individuales de las uniones, los cuales dependen de la diferencia entre las temperaturas de las uniones.

Para medir la diferencia de temperatura, solamente es necesario abrir la malla en un punto conveniente (en algún punto frío) el insertar un voltmetro que debe ser bastante sensible dado que el voltaje producido por un termopar está en el rango de los milivolts como se muestra en la siguiente figura.

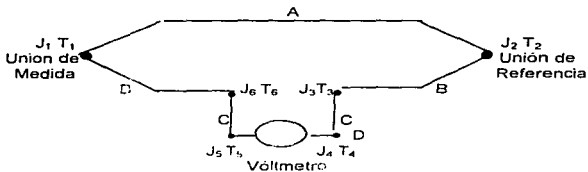


Fig. 5.3 Figura que nos ilustra como medir diferencias de temperatura

Cuando se inserta un voltmetro en la malla de un Termopar, generalmente es más conveniente insertarlo como lo mostramos en la figura anterior. En dicha figura, el metal A y el metal B no se tocan uno con otro en el punto de la unión fría. En lugar de esto ambos están en contacto con conductores de cobre. El voltaje neto en la malla permanece invariable dado que ahora hay dos uniones frías, una entre el metal A y el cobre y la otra entre el metal B y el cobre. La suma de los dos voltajes de unión producidos por estas dos uniones frías es igual al voltaje que se producirá por la sola unión fría de los metales A y B. Desde luego, las dos uniones frías deben mantenerse a la misma temperatura que experimentaría la unión simple.

Para evitar el problema de identificar los termopares por el nombre de fábrica, se ha adoptado un código de una letra para los tipos de termopares mas comúnmente usados. De esta forma, un termopar J está constituido de Hierro y Constanta, el E es Chromel y Constanta, etc. En cada caso el primer metal o aleación metálica mencionada es el lado positivo, mientras que el segundo material o aleación metálica es el lado negativo.

5.3.2.1 FUENTES DE ERROR EN LOS TERMOPARES.

UNION ABIERTA: Hay muchas formas para que una unión se abra. El error inducido por una unión abierta es de tal magnitud que es facil de identificar. Con la simple medida de la resistencia del termopar, se puede identificar si hay una punta abierta.

DESCALIBRACION: Este error es una falla muy seria, el cual puede causar más errores que no podrían ser detectados. La descalibración se debe a alteraciones en las características de los materiales del Termopar, esto cambia el voltaje Seebeck. Puede ser causado por exponer los alambres a temperaturas excesivamente altas, partículas contaminantes dentro de los alambres, o por trabajar los alambres en frío. El último efecto puede causarse por un fuerte estiramiento a lo largo del conductor.

DEGRADACION DEL AISLAMIENTO: En algunos casos el aislamiento puede desintegrarse y causar una filtración de resistencia que puede provocar un error en la medida del voltaje Seebeck, en suma, agentes químicos en el aislamiento pueden introducirse hasta los alambres provocando una descalibración.

ACCION GALVANICA: Al entrar en contacto sustancias químicas con los alambres del Termopar pueden provocar un efecto galvánico. Resultando de esto un voltaje que puede ser más de 100 veces mayor que el efecto Seebeck, provocando errores extremos.

CONDICIONES TERMICAS: Los termopares pueden desviar energía calorífica lejos de la fuente que será medida. Para medir pequeñas masas, podemos usar termopares con alambres de diámetro pequeño. Sin embargo, los alambres de diámetro pequeño son más susceptibles a los efectos descritos previamente.

5.4 TERMISTORES.

Un termistor es una resistencia sensible a la temperatura, esto es, un cuerpo que muestra cambios en su resistencia eléctrica al modificarse la temperatura del cuerpo. A diferencia de los RTD's, los cuales consisten de alambres enrollados o películas de metal que muestran pequeños cambios de resistencia con la temperatura, los termistores son semiconductores cerámicos que muestran grandes cambios en su resistencia con la temperatura.

Existen termistores con coeficiente de temperatura (-) y (+), pero los primeros son usados muy rara vez para medir temperatura, actualmente son de mayor interés y uso para medir la temperatura los termistores compuestos por óxido de metales (Magnesio, Cobalto, Niquel, Cobre, Hierro y Titanio).

Una ventaja de los termistores disponibles es el amplio rango de valores de resistencia. El uso de los termistores de alta resistencia hace posible obtener medidas a distancia a través de dos cables sin necesidad de compensar la temperatura de los cables porque su resistencia es insignificante.

Virtualmente no hay límite en la longitud del cable que puede ser usado entre el sensor y el instrumental asociado, cuando es hecha una adecuada elección del termistor.

5.4.1 CARACTERÍSTICAS DE RESISTENCIA - TEMPERATURA DE TERMISTORES Y RTDS

Estas ventajas se obtienen de la no linealidad y del relativamente estrecho rango de operación para cualquier termistor especificado. Por otro lado, los RTD's de Platino son lineales en un amplio rango.

Las ventajas adicionales presentadas por los termistores, es la posibilidad de presentarlos en pequeños tamaños 0.05" de diámetro y un tiempo de respuesta más rápido.

Los Termistores son resistencias sensibles a la temperatura hechos de una mezcla de Oxidos Metálicos. Los termistores para medir temperatura tienen un amplio coeficiente (negativo) de resistencia a la temperatura, que va desde el 2% hasta el 6%/°C, esta alta sensibilidad hace a los termistores especialmente útiles donde las medidas de alta resolución "0.01C" son necesarias en un rango limitado de temperatura. Existe una amplia variedad de formas y tamaños de termistores; cuentas, discos, arandelas, barras, cilindros de prueba, estas son configuraciones típicas.

Los elementos en miniatura son buenos para aplicaciones donde la respuesta es muy rápida y se desea una cantidad térmica mínima.

En la actualidad los termistores herméticamente sellados tienen una excelente estabilidad a largo plazo, con tal de que ellos no sean expuestos a temperaturas altas durante mucho tiempo. La estabilidad de la mayoría de los termistores se degrada arriba de 300°C.

Cuando son usados en conjunto con un sistema de adquisición de datos, la tarea de la linealización puede ser frecuentemente simplificada por sistemas de control o cálculos algebraicos.

5.4.2 TECNICAS DE FABRICACION DE LOS TERMISTORES.

Los Termistores comerciales pueden ser agrupados en dos grandes clases de acuerdo con el método por el cual sus electrodos son conectados. Cada una de estas puede ser subdividida en varios tipos caracterizados por diferencias en su geometría, en sus técnicas de procesamiento y en su encapsulado. Las clases y tipos más comúnmente usados son mostrados en las tablas siguientes:

UNIDADES CON SALIDAS EMPOTRADAS DENTRO DEL CUERPO DEL TERMISTOR.	UNIDADES CON CONTACTOS METÁLICOS APLICADOS A LA SUPERFICIE DEL TERMISTOR.
Cuentas descubiertas	Discos
Cuentas con cubierta de vidrio	Chips
Sonda de vidrio	Hojuela
Varilla de vidrio	Varillas
Cuentas de vidrio	Arandelas
Sondas con salidas axiales	Obleas

Tabla 5.2 Clases en las que se pueden agrupar los termistores.

CARACTERÍSTICAS DE SONDAS Y CUENTAS

VENTAJAS	DESVENTAJAS
1.- Alta estabilidad y confiabilidad	1 - Relativamente alto costo para tolerancias e intercambiabilidad
2.- Disponible en tamaños muy pequeños (0.1 mm. de diámetro)	2 - Requerimientos de intercambiabilidad de resistores en la red
3.- Respuesta rápida.	3 - Constantes de disipación comparativamente bajas
4.- Puedo ser usado a altas temperaturas (300 °C)	

Tabla 5.3 Características de los tipo sondas y cuentas.

CARACTERÍSTICAS DE DISCOS OBLEAS Y CHIPS.

VENTAJAS.	DESVENTAJAS.
1.- Tolerancia e intercambiabilidad en costo relativamente bajo	1 - Dificultad para obtener electrodos confiables
2.- Constantes de disipación relativamente altas.	2 - Licencia para estabilidad y confiabilidad mucho más deseada
3.- Respuesta rápida razonable en tiempos para unidades descubiertas particularmente para 1X1 mm X 1/8 de espesor descubierto, y cerca de 1.5 mm de diámetro cubierto	3 - Tamaño limitado hasta cerca de 1 mm. de diámetro
	4 - no puede ser usado a altas temperaturas Requiere esterilización fría
	5 - Respuesta comparativamente larga para unidades cubiertas

Tabla 5.4 Características de discos obleas y chips.

Hoy en día la mayoría de los termistores son fabricados utilizando tecnología cerámica. El proceso básico es aquel en el cual se mezclan dos o mas polvos de Oxidos semiconductores y un aditivo conveniente. Al variar los tipos de óxidos usados y sus proporciones relativas un amplio rango de valores de resistencia y de valores característicos pueden ser obtenidos.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

TRANSDUCTORES Y SENSORES.

5.4.3 TIPOS DE TERMISTORES EXISTENTES.

TIPOS DE TERMISTORES EXISTENTES.	
TIPO	CARACTERÍSTICAS.
cuenta	Los tipos más comúnmente usados son la cuenta revestida de cristal y el cristal de prueba. Las cuentas revestidas de cristal están disponibles en diámetros desde 0.125 mm hasta 1.5 mm. Las de cristal de prueba están disponibles desde 1.5 mm hasta 6.35 mm o en diámetros desde 0.4 mm hasta 2.5 mm.
disco	Los termistores tipo disco que tienen diámetros desde 1 mm hasta 3 mm son usados frecuentemente para termómetros con un costo bajo.
chip o galleta	Los termistores tipo chip o galleta son fabricados frecuentemente usando un proceso de grabado en el que una parte del material es colocado sobre una banda que lo lleva hasta una hoja fija. Están disponibles con secciones transversales desde 1 mm X 1 mm hasta 13 mm X 13 mm y en espesores desde 0.25 mm hasta 1.5 mm.
escama u oblea.	Los termistores tipo oblea se fabrican usando alguna de las variadas técnicas usadas en la fabricación de capacitores y resistencias de película gruesa. Están disponibles en espesores desde 0.025 hasta 0.05 mm y sección transversal de 0.5 X 0.5 mm a 3 X 3 mm.
barra	Los termistores tipo barra están hechos de la obtención de una mezcla de polvos de óxido. Son hechos en diámetros de 0.5 mm hasta 50 mm. Si bien una gran cantidad de barras de diámetro pequeño son usadas para medir la temperatura en la atmósfera en radio sondas, los termistores tipo barra no son tan usados en los termómetros de resistencia como lo son los tipo cuenta o tipo disco.
rondana	Los termistores tipo rondana están hechos de manera similar que los tipo disco excepto por que a estos se les forma un agujero en el centro durante el proceso de comprimido. Además estas unidades son usadas rara vez para termómetros de resistencia.

Tabla 5.5 Tipos y características de termistores existentes.

5.4.4 PRUEBA Y CALIBRACION DE TERMISTORES.

Los termistores son calibrados usualmente en una tina con líquido que debe tener las siguientes características: 1.) Baja conductividad eléctrica; 2.) Baja Viscosidad; 3.) Alta conductividad Térmica.

Un puente de Wheatstone detector de cero preciso y sensible es un detector recomendado para medir la resistencia. El voltaje a través del puente deberá mantenerse bajo para eliminar errores de autocalentamiento. En general si se sabe que el detector de cero tiene una sensibilidad adecuada, la técnica siguiente podrá ser usada para determinar el voltaje puente:

- 1) Poner el voltaje puente a un valor muy bajo y balancear el puente
- 2) Desbalancear el puente por medio de una pequeña reducción de las resistencias colocadas en el puente.
- 3) Un lento incremento del voltaje puente
- 4) Si el indicador del detector de cero se mueve hacia una condición de desbalanceo el voltaje puente será fijado demasiado alto. Si aumenta el desbalanceo, el incremento de voltaje simplemente dará un incremento de sensibilidad
- 5) Continuar incrementando lentamente el voltaje puente hasta que la condición de desbalanceo deje de crecer e inicie un proceso inverso
- 6) Disminuir ligeramente el voltaje puente, ahora habremos establecido la condición mínima de autocalentamiento
- 7) Rebalancear el puente y determinar si el cambio mínimo detectable en la resistencia es adecuado para la incertidumbre deseada en la temperatura

La precisión del puente de Wheatstone debe también ser consecuente con la precisión de los medidores de temperatura finalmente requerida por el termistor. Es recomendable que la precisión de un circuito puente sea de 0.01%/°C

La sensibilidad en las medidas a menudo puede ser incrementada, cuando usamos un puente de Wheatstone y un detector de cero, por la aplicación del voltaje puente en forma de pequeños pulsos, teniendo un ciclo de trabajo pequeño. De esta manera la magnitud del voltaje puente puede incrementarse sin un incremento correspondiente en el promedio de potencia disipada por el termistor.

- 1) En la unión de los alambres del termistor y la textura fija
- 2) Dependiendo en la construcción de la prueba de fijado dentro de la fijación

Los valores de resistencia de los termistores usados para termómetros de resistencia generalmente están dentro del rango de 100 a 100 kohms, la constante de disipación está en el rango de 0.1 a 5 mW/°C.

5.4.5 RETRASO EN LA RESPUESTA O RETARDO.

Tal vez los grandes problemas asociados con la calibración de termistores de respuesta rápida es el retardo térmico que puede existir entre el monitor de temperatura y el termistor en prueba. Una técnica estandar usada por los fabricantes para calibrar termistores estandar

contra un termómetro de resistencia de platino, es usando los termistores como monitor. Cuando tal calibración es realizada, es necesario monitorear el lote con el termistor de prueba, durante la calibración, para asegurar un adecuado control de lote. Cuando la constante de tiempo del termistor difiere significativamente de la del termómetro de resistencia de platino, es necesario mantener el grupo de fluctuaciones (indicadas por el termistor) hasta que la calibración sea exactamente la requerida. Esto hará progresivamente más fácil que el tiempo de respuesta del termistor se reduzca. En este caso, es deseado el uso de un bloque metálico de integración térmica. En general, es deseable para el monitoreo de temperatura tener un tiempo de respuesta comparable al de los termistores de prueba.

Una técnica que puede facilitar el requerimiento de un control cerrado de temperatura es usando un puente de comparación en donde una de las ramas es reemplazada por un termistor que tiene el mismo tiempo de respuesta y valor que el termistor de prueba.

5.4.6 ESTABILIDAD DE TERMISTORES.

La estabilidad de un termistor es una medida de cambio que ocurre en un tiempo dado o como resultado directo de una o más exposiciones a ambientes específicos. En general, los parámetros de mayor interés son: 1) La Resistividad, 2) la constante de disipación asociada al material semiconductor y 3) La resistencia de contacto entre el semiconductor y los electrodos (que además afectan el tiempo termal y la constante de disipación).

Cambios en las propiedades térmicas (constante termal y constante de disipación) de un termistor son raras. Tales cambios es más probable que ocurran en termistores de prueba ensamblados para variaciones en transferencia de calor entre el elemento termistor y el medio ambiente.

Cambios significativos en la constante de disipación "b" son relativamente poco frecuentes, salvo que los termistores sean expuestos a temperaturas excesivamente altas. Desde luego, para termistores intercambiables pequeños cambios en "b" pueden ser significativos. Los parámetros que están sujetos a cambios son la resistividad del semiconductor y la resistencia de contacto.

5.4.7 LINEALIDAD DE TERMISTORES.

Hay una gran variedad de circuitos en que un termistor puede ser usado para medir temperatura. En muchos casos, un criterio mayor es que el circuito otorgue una salida que sea lineal con la temperatura. Cuando se desea el uso de una fuente de corriente constante, el circuito usado deberá ser una malla de dos terminales que presente una resistencia lineal contra características de temperatura. Bajo estas condiciones, un voltmetro digital conectado a la malla puede desplegar directamente temperatura cuando es seleccionada la combinación adecuada de niveles de corriente y resistencia. Si por otro lado, el uso de una fuente de voltaje

constante es conveniente, el circuito usado será una malla de dos terminales que presente una conductancia lineal contra características de temperatura.

Como ejemplo, una red usada con una fuente de voltaje constante es el divisor de voltaje mostrado en la siguiente figura.

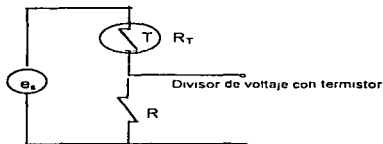


Fig 5.4 Linealización de un Termistor

El voltaje de salida para este circuito es generalmente tomado desde la resistencia fija, entonces el efecto de la carga es invariante con la temperatura para este arreglo. Por lo tanto, por el análisis propuesto, la resistencia R puede ser considerada como el paralelo de la carga y el divisor de resistencias. El voltaje de salida también se aumenta con incrementos de temperatura para este arreglo.

5.5 TRANSDUCTORES Y SENSORES DE PRESION.

Un sensor de presión puede definirse de varias formas equivalentes. Una definición común dice que es el vínculo entre una fuente de presión y un circuito electrónico. Nosotros definimos un sensor de presión como un dispositivo que forma parte del control de un proceso que sensa un cambio de presión en un medio y que provoca cambios en un circuito electrónico [21].

Los sensores de presión están disponibles en una amplia variedad de rangos de presión, razones eléctricas, construcción, materiales, exactitudes y costos. Pueden usarse para sensar presiones desde el alto vacío hasta 50,000 PSI o mayores y pueden operarse con fluidos desde agua o aire hasta líquidos corrosivos o gases y materiales radioactivos.

Un cambio en presión es usualmente sentido y detectado en una de dos formas básicas; el movimiento o el desplazamiento de un objeto bajo un incremento de la fuerza como puede verse en la carrera de un pistón en un cilindro, o el cambio de alguna propiedad física como la

resistencia eléctrica de un material debido a incrementos de la presión interna provocados por el aumento de la fuerza aplicada.

Otra posibilidad incluye el uso de efectos piezoeléctricos, cambios del índice de refracción con respecto a la presión. Estos efectos son usados frecuentemente en sensores con un costo elevado y tienen un rango limitado en su respuesta.

Casi todos los sensores de presión caen dentro de tres categorías; dependiendo de la presión que será sensada. Todos los sensores de presión responden a una diferencia entre la presión sensada y una presión de referencia conocida.

5.5.1 SENSORES DE PRESION ABSOLUTA.

Un sensor de presión absoluta está diseñado para sensar un cambio en presión absoluta independiente de la presión atmosférica prevaleciente. Un lado del elemento sensor es expuesto a la fuente de presión, y el otro lado es expuesto a un muy alto vacío, donde la presión de referencia es cero. Un sensor de presión absoluta es útil cuando la presión a sensar es menor que la atmosférica o cuando la presión de referencia puede cambiar, tal como ocurriría en un cambio de altitud.

5.5.2 SENSORES DE PRESION GAGE.

Un sensor de presión Gage está diseñado para sensar presión arriba de la atmosférica. Un lado del sensor es expuesto a la fuente de presión, y el otro lado está abierto a la presión atmosférica, en este caso la presión de referencia es la de la atmósfera. El sensor de presión Gage es probablemente el tipo más comúnmente usado.

5.5.3 SENSOR DE PRESION DIFERENCIAL.

Este sensor por su nombre se supone que responde a la diferencia de dos presiones. En este caso, la presión de referencia es una de las presiones variables. Físicamente este sensor puede ser accionado directamente o por una combinación de dos presiones absolutas o Gage.

5.5.4 TRANSDUCTOR DE PRESION INDUCTIVO.

El transductor de presión inductivo usa la presión para mover una membrana mecánica que es usada para cambiar la inductancia de una bobina. La inductancia es una función del movimiento relativo de un núcleo y la bobina inductiva. Los Transductores de bobina inductiva son usados a veces en circuitos osciladores, para centrar la frecuencia de oscilación.

Transductores de una bobina tienen problemas de compensación para efectos de temperatura. Esto requiere de un arreglo del núcleo y la bobina para cambios de temperatura y permeabilidad.

Hay un tipo de transductor inductivo que usa la relación de la reluctancia de dos bobinas. Este transductor de reluctancia es menos sensible a efectos de temperatura que el de una bobina. Un movimiento de aproximadamente 0.003" produce una salida de voltaje de AC de aproximadamente 100mV.

En un transductor de presión reluctivo tipo diafragma, un diafragma de material magnético es colocado entre dos inductancias ensambladas simétricamente. Este diafragma es desviado cuando hay una diferencia de presión entre las dos puertas de entrada. El hueco en el flujo magnético hacia uno de los núcleos se incrementa; en el otro núcleo decrece. En esta forma la reluctancia cambia con la presión. El efecto neto es un cambio en la inductancia de las dos bobinas del transductor. La relación de inductancias L_1/L_2 (L_1 inductancia de la bobina 1 y L_2 inductancia de la bobina 2), puede ser medida con un circuito puente para detectar el voltaje proporcional a la diferencia de presión. Medir la fuerza debido a la presión externa de este modo es complicado por el cambio en la inductancia de la relación de bobinas. La fuerza será medida cambiando el acoplamiento magnético del transductor como el resultado del desplazamiento del núcleo. Los errores de histéresis en este tipo de transductores son limitados por los componentes mecánicos. La forma de la fuerza sumando miembros está gobernada por esas configuraciones que están mejor adaptadas para operar con el principio inductivo.

El diafragma puede ser usado también como una parte del lazo inductivo. Aquí el comportamiento total del transductor puede ser mejorado ya que las características mecánicas del diafragma deben ser casi óptimas para mejorar las características magnéticas.

Estos tipos de transductores proporciona las siguientes ventajas y desventajas:

VENTAJAS	DESVENTAJAS.
1.- Es posible tener una salida elevada	1.- La respuesta en frecuencia es normalmente limitada por sus componentes mecánicos.
2.- Puede responder a medidores dinámicos y estáticos	2.- La excitación debe ser de corriente alterna
3.- Da una resolución continua	3.- El transductor debe estar reluctiva y resistivamente balanceado a cero
4.- Es posible una alta relación señal a ruido.	4.- Objetos y campos magnéticos cercanos pueden causar errores transductor
5.- puede proporcionar directamente frecuencias moduladas para circuitos de telemetría	5.- El desplazamiento volumétrico tiende a ser grande
	6.- La fricción mecánica puede causar desgaste y error en un periodo de tiempo.

Tabla 5.5 Ventajas y desventajas de los transductores proporcionales de presión

5.5.5 TRANSDUCTORES DE PRESION PIEZOELECTRICOS.

El transductor de presión piezoeléctrico usa un cristal; para generar una carga o voltaje cuando es presionado mecánicamente. Un diafragma normalmente reacciona con la presión que produce la carga en el cristal. Un transductor de presión piezoeléctrico puede operar sobre un amplio rango de temperatura con errores relativamente pequeños. Algunas unidades de cuarzo usan cristales seleccionados de aquellos encontrados en estado natural. Los cristales desarrollados incluyendo el ADP (Ammonium dihydrogen phosphate), son desarrollados en una solución acuosa. Varios materiales cerámicos son también usados.

Los elementos piezoeléctricos son cortados del cristal a lo largo de ejes cristalográficos existentes. Elementos cerámicos son comprimidos de polvos de materiales en la forma requerida y entonces homeados a altas temperaturas. Las características piezoeléctricas son el resultado de como estos son polarizados por la duración de un campo eléctrico durante el proceso de enfriamiento.

5.6 CONSIDERACIONES.

Es importante aclarar que, en el sistema estamos utilizando potenciómetros para simular la variación de voltaje que nos proporcionarán los sensores, los cuales deben ser colocado en cada una de las entradas al controlador.

También las salidas están siendo simuladas de tal manera que cero represente el estado apagado y uno el estado encendido de los actuadores del sistema.

CAPITULO 6.

PRUEBAS Y RESULTADOS

6.1 INTRODUCCIÓN.

Este capítulo tiene la finalidad de demostrar a través de los resultados obtenidos durante la etapa de pruebas lo confiable que son los algoritmos de control difuso (si lo comparamos con los tradicionales), así como la implementación de éstos en una arquitectura de tipo pipeline.

En los siguientes párrafos damos una recopilación de las pruebas que se hicieron, a los algoritmos y a la arquitectura por separado y en conjunto.

6.2 PRUEBAS.

6..2.1 ALGORITMOS.

Los algoritmos de control difuso se diseñaron de tal manera que existe una relación entre cada par de variables (presión y temperatura), para realizar los procesos de fusificación y defusificación, obteniendo como resultado dos valores de defusificación, uno para presión y otro para temperatura. Las pruebas que se realizaron a los algoritmos fueron el proceso de fusificación y defusificación de las variables.

Como se puede ver al analizar las figuras del capítulo 3 (proceso de fusificación y defusificación de un punto), los resultados obtenidos de las pruebas a los algoritmos nos permiten decir que cubren las necesidades de nuestro proceso, con respecto a la exactitud de resultados, ya que para la temperatura obtuvimos una diferencia de 3 grados con respecto al valor exacto y para la presión el valor difirió en 1 psi del valor exacto, además si hacemos un análisis del tiempo de procesamiento nos damos cuenta que este es muy bajo, por lo tanto, basándonos en las justificaciones anteriores, tomamos la decisión de utilizar los algoritmos diseñados en la implementación del sistema de control con la certeza de que funcionarían en forma eficiente.

6.2.2 ARQUITECTURA.

Las pruebas realizadas en la arquitectura, fueron las siguientes.

68HC11F1	68HC11A8	INTEL 8088.
-Lectura de a cámara del C A/D	-Recepción de datos.	-Recepción de datos.
-Proceso de fusificación	-Evaluación de reglas	-Proceso de autosintonización
-Primera parte de la defusificación (Obtención de mínimos)	-Segunda parte de la defusificación (Método del centroide)	-Manipulación de actuadores
	-Transmisión de datos.	-Servicio de interrupción para transmisión de datos a la PC

Fig. 6.3 Tabla de las pruebas realizadas a las arquitecturas

El aspecto importante de estas pruebas es el relacionado con el tiempo que tarda cada uno de los microprocesadores en realizar sus procesos y el tiempo total de una muestra. Dos temperaturas y dos presiones- así como el número de muestras que toma nuestro sistema en un segundo, además de comprobar ya en forma de programas la eficiencia de los algoritmos.

En la siguiente tabla se enlistan los tiempos obtenidos en las pruebas.

Proceso	68HC11F1	68HC11A8	8088	Total
Lectura del convertidor A/D	17.00×10^{-6} s			17.00×10^{-6} s
Proceso de fusificación	17.24×10^{-6} s			17.24×10^{-6} s
1ª parte de defusificación	7.83×10^{-6} s			7.83×10^{-6} s
Transmisión de datos	0.87×10^{-6} s	24.00×10^{-6} s		8.97×10^{-6} s
Recepción de datos		847.00×10^{-6} s	42.00×10^{-6} s	889.00×10^{-6} s
Evaluación de reglas		9.82×10^{-6} s		9.82×10^{-6} s
2ª parte de defusificación		2.38×10^{-6} s		2.38×10^{-6} s
Autosintonización			8.07×10^{-6} s	8.07×10^{-6} s
Manipulación de actuadores			13.00×10^{-6} s	13.00×10^{-6} s
TOTALES:	0.02596 seg	0.002091 seg	0.00062 seg	0.0289 seg

Tabla 6.4 Tiempo utilizado durante las pruebas para cada etapa del algoritmo.

De la tabla anterior podemos ver que el proceso completo para cada uno de los datos tarda un promedio de **0.0289** segundos (29 mseg. aproximadamente), por lo que una muestra de cuatro datos (dos temperaturas y dos presiones), tardaría en procesarse un promedio de **0.1156** segundos lo que nos da al final 8 muestras por segundo aproximadamente.

También de la tabla podemos obtener el tiempo en que se realiza cada uno de los procesos (Fusificación, Defusificación, Autosintonización y Manipulación de válvulas), incluyendo la comunicación con la PC, que nos da como resultado el tiempo total de procesamiento; desde la toma de la muestra hasta la transmisión de datos por el puerto serial a la PC, si ese fuera el caso.

PROCESO	TIEMPO DE EJECUCIÓN	
1) Fusificación		0.07252 seg.
2) Defusificación	+	0.03968 seg.
3) Autosintonización y Manipulación de válvulas	+	0.00344 seg.
4) Transmisión de datos a la PC	+	0.3017x10 ⁻³ seg.
TOTAL:		0.11594 seg.

Fig. 6.4 Proceso y tiempos de ejecución

6.3 RESULTADOS

En nuestro sistema los resultados son los valores que se obtuvieron del proceso de fusificación y defusificación de cada una de las variables de entrada. Tomando en cuenta que este proceso se realiza por pares, en la siguiente tabla se presenta una muestra de las combinaciones de valores, su defusificación y su valor de defusificación obtenido del sistema, para hacer al final una comparación de ambos.

Combinación de datos		Valor de Defusificación (papel)		Valor de Defusificación (sistema)	
Temp °C	Pres. (psi)	Temp	Pres.	Temp	Pres.
22	31	70	20	70	20
80	32	70	20	70	20
90	33	71	20	70	21
100	34	72	21	70	21
110	35	72	21	70	21
130	36	7C	23	7A	22
135	37	7D	23	7A	23
140	38	8E	24	8E	24
145	39	8E	24	8E	24
120	30	75	22	78	22
127	32	7F	22	7F	22
127	39	7F	24	7F	24
127	35	7F	24	7F	24
130	30	8E	22	8E	22
130	32	8E	23	8E	24

Haciendo un análisis de la tabla anterior podemos concluir que el sistema realiza el proceso de fusificación y defusificación de forma casi exacta, teniendo pequeñas diferencias en algunos puntos, sin ser estas causa de la desaprobación de los algoritmos de control, ya que con la etapa de autosintonización esto se corrige, quedando eliminado totalmente el error que se presenta en algunos puntos al momento de defusificar.

Finalmente al realizar un análisis global del capítulo, que implica un análisis del sistema, podemos ver que la implementación del algoritmo de control difuso en una arquitectura como la que utilizamos, nos permitió durante las pruebas ir verificando por etapas los resultados de los diferentes procesos realizados y por lo tanto asegurarnos de que partes estaban funcionando y cuales había que mejorar. Dándonos como resultado final un proceso con una exactitud y tiempo de procesamiento adecuados para satisfacer las necesidades de nuestro proceso.

6.4 MANEJO DEL PROCESO DESDE EL MONITOR CENTRAL.

Nuestro sistema puede formar parte de un sistema de control distribuido [23], monitoreado desde un monitor central (PC) [22]. El cual cuenta con un software especial (sistema operativo QNX multitareas semejante al UNIX), que nos permite estar sensando las variables de entrada de los diferentes controladores que este monitoreando.

El monitor central esta interconectado con nuestro controlador mediante el puerto serie, y es capaz de realizar las siguientes actividades:

a) **MONITOREAR LOS VALORES QUE TIENEN LAS VARIABLES DE ENTRADA:** Esto se realiza mediante el uso de códigos que el monitor central le envía al controlador cada vez que el primero desea actualizar la información que tiene. El controlador interrumpe su proceso para atender la petición del monitor central. Al identificar el comando que le fue enviado, contesta con la información solicitada por la PC terminando así, por el momento la comunicación con el monitor central (PC), retomando el proceso en la instrucción que se encontraba ejecutando.

b) **CONTROL DE ACTUADORES DESDE EL MONITOR CENTRAL:** Esta acción consiste en que el operador enviará desde el monitor central (PC), el código que le permita tomar el control del sistema. En el instante en que el controlador identifica este código interrumpe su proceso, contestando con un eco al monitor central. A partir de este momento la manipulación de los actuadores de salida del sistema la ejecuta la PC, ya que cuenta con comandos que le permiten activar o desactivar los actuadores sin que el algoritmo de control necesite estar activado, reemplazando con esto al controlador e introduciendo la experiencia de un experto (operador del sistema), que es quien maneja el monitor central, quedando el controlador simplemente como una interface de recepción y envío de los valores que en ese

instante tengan las variables de entrada al sistema. Si se desea devolver el control al controlador, también existe un código con el que se puede lograr, activando el algoritmo de control para retomar el control del sistema, a partir de las condiciones que nos haya dejado el experto.

Los códigos que manejamos para que el monitor central pueda pedir información y/o tomar el control del sistema son los siguientes:

CODIGO	DESCRIPCION
06	- Pide información la PC al controlador
08	- La PC desactiva algoritmo de control
12	- La PC activa algoritmo de control
42	- La PC activa actuador para T ₁
43	- La PC activa actuador para P ₁
44	- La PC activa actuador para T ₂
45	- La PC activa actuador para P ₂
54	- La PC desactiva actuador de T ₁
55	- La PC desactiva actuador de P ₁
56	- La PC desactiva actuador de T ₂
57	- La PC desactiva actuador de P ₂

Tabla 6.5 Codigos utilizados para interactuar con el minitor central.

CAPITULO 7

CONCLUSIONES.

Como todos, al inicio de la tesis teníamos la incertidumbre de que tan capaces éramos para poder cumplir los objetivos del trabajo. Esto debido a lo amplio que son los temas que se involucran en nuestra tesis. Pero al final logramos cubrirlos totalmente, obteniendo con esto un complemento a la formación adquirida durante nuestra carrera y una gran satisfacción personal.

Este capítulo contiene las conclusiones a las que llegamos al final de nuestro trabajo. Pensamos que debido a lo amplio de este y a la variedad de temas involucrados en él, para tener una mayor claridad en nuestras conclusiones, era conveniente dividirlos en tres bloques principales que están incluidos en nuestra tesis. Estos son lógica difusa y algoritmos de control, procesamiento paralelo y microprocesadores y/o microcontroladores. Es importante hacer notar que el orden en que los describamos no implica que uno sea más importante que el otro, ya que para nosotros los tres tuvieron y tienen la misma importancia y fueron fundamentales en nuestro trabajo.

Si empezamos por el procesamiento paralelo, podemos concluir que es una arma muy poderosa para satisfacer las demandas en cuanto a velocidad de procesamiento y ahorro de memoria se refiere. Y que debido a la gran cantidad de información que los sistemas de cómputo actuales manejan, esta es una técnica bastante poderosa para satisfacer a los usuarios, que cada día demandan mayor eficiencia en el manejo de su información.

Es un tema bastante amplio, digno de ser atendido con mayor interés dentro de nuestra facultad. En la actualidad sus aplicaciones son muy amplias y no dudamos que en un futuro todas las personas relacionadas con el diseño y fabricación de sistemas de cómputo tengan la necesidad de tomar este rumbo para mejorar sus servicios.

En cuanto a nuestro trabajo se refiere, el haber aplicado este tipo de procesamiento nos dio la pauta para desarrollar un sistema más eficiente, ya que nos proporcionó la opción de dividir nuestro algoritmo en tareas más pequeñas y por lo tanto más fácil de atender, lo cual nos facilitó el desarrollo de nuestros programas, que al tratarse de algoritmos de control, por lo general son demasiado difícil de programar.

En relación al Hardware este procesamiento nos permite contar con una arquitectura más sofisticada y poderosa, que de ser necesario tiene la opción de interconectarse con otros procesadores para formar una red que pueda manejar un gran número de variables, con una alta velocidad de procesamiento y eficiente administración de memoria.

En lo que a Software se refiere el desarrollo de este tipo de procesamiento ha venido a darle un gran avance, sin que esto implique mayor dificultad, si no al contrario, ya que los programas se pueden dividir en tareas más pequeñas, lo que hace más fácil su ejecución, además

pueden ser atendidas de forma simultánea por diferentes procesadores, obteniendo con esto un ahorro en el tiempo de procesamiento y una mejor administración de memoria.

Los conceptos anteriores nos sirvieron como base en el desarrollo de nuestra tesis, ya que uno de los objetivos era lograr que el sistema pudiera realizar las tareas asignadas a cada etapa en el menor tiempo posible y ocupando el menor espacio en memoria; otro objetivo que debía cumplir nuestro sistema es que pudiera interactuar con otros sistemas de control a través de un dispositivo de monitoreo central y que además estuviera presenta la opción de interconectarlo con otros dispositivos para formar una red de control que pueda manejar varias variables. La única forma de lograrlo era con una arquitectura de Procesamiento Paralelo

No fue fácil decidir cual de los tipos de procesamiento era el más apropiado para cubrir las necesidades de proyecto, primero por que cualquiera de ellos es capaz de cumplir las tareas del algoritmo en forma eficiente y después por que involucrarse en sus diseños es en verdad interesante y nos deja bastantes conocimientos. Al final decidimos usar una Arquitectura tipo "pipeline", debido a que su estructura se apegaba por completo a la de nuestro algoritmo, además de que durante la etapa de pruebas trabajo en forma eficiente, dándonos muy buenos resultados.

En lo que a Lógica Difusa aplicada a sistemas de control se refiere, no contábamos con ningún antecedente académico que nos permitiera valorar su aplicación. Por lo tanto, nuestra decisión de usarla para el desarrollo del sistema de control, se basa en lo fácil que resulta el diseño de algoritmos bajo esta lógica, ya que algo que hace demasiado complicados a los algoritmos de control tradicionales es el modelo matemático que los representa, cosa que no es necesario conocer para el diseño de un algoritmo difuso, lo cual los hace fáciles de programar, además de que ocupan muy poco espacio en memoria.

Podemos concluir que la Lógica Difusa es un método bastante eficiente para el desarrollo de algoritmos de control, y que para sistemas de control muy complicados es una herramienta ideal ya que el pasar el algoritmo a algún lenguaje de programación es demasiado fácil.

Usar esta técnica nos dejó completamente satisfechos, ya que aquellas clases de control tan complicadas por el nivel de matemáticas que había que manejar para la solución de sistemas, bajo esta lógica se vuelven procedimientos, incluso hasta, agradables y fáciles de entender. Ya que el manejo de las variables de entrada como variables lingüísticas hace muy entendible un problema y por lo tanto fácil de resolver.

Con relación a microcontroladores y microprocesadores, habíamos manejado algunos durante la carrera, por lo que nos pareció muy interesante trabajar con el 68HC11 y el 8088 en el diseño del sistema, ya que cuentan con las características apropiadas para un sistema de control: (CA/D, Timers, Interrupciones, Puertos paralelos y Puerto serie, instrucciones. muy poderosas, etc.). Cuentan con la característica de tener arquitecturas pequeñas y por lo tanto fácil y rápido de construir, la velocidad a la que operan es apropiada para nuestra aplicación, se adaptaron fácilmente a arquitecturas de Procesamiento Paralelo, por lo que en general los consideramos adecuados para la aplicación que les dimos.

Durante el desarrollo de nuestro sistema nos encontramos con algunos problemas para comunicar en forma eficiente al HC11 con el 8088, lo cual nos obligó a insertar rutinas de

tiempo en el momento de la comunicación para asegurarnos que no existiera ningún problema en el bus de datos.

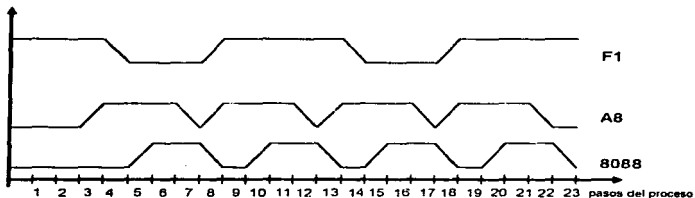
ANEXOS

Esta parte de la Tesis contiene gráficas de diagramas de bloques, diagramas de flujo, diagrama de estados para cada microprocesador y lista de tareas por estado, las principales características eléctricas de los microprocesadores que estamos usando y los diagramas eléctricos de las diferentes etapas de la arquitectura.

ANEXO 1

DIAGRAMA DE TIEMPOS PARA CADA MICROPROCESADOR Y LISTA DE TAREAS EJECUTADAS.

A continuación se dan las gráficas a cerca del funcionamiento de la arquitectura a manera de bloques separados de acuerdo al módulo que se trata, así como también viene acompañadas de una breve explicación. La primera corresponde al 68HC11F1, la segunda al 68HC11A8 y la tercera al 8088.



- 1) F1) Realiza la búsqueda de los datos para fusificación
 A8) Pregunta si F1 si ya tiene un dato
 88) Pregunta si el A8 tiene un dato
- 2) F1) Realiza la determinación de los mínimos
 A8) Pregunta si F1 si ya tiene un dato
 88) -Pregunta si el A8 tiene un dato
- 3) F1) -Avisa que tiene datos de fusificación para T_i
 A8) -Identifica que ya hay un dato a la entrada
 88) -Pregunta si el A8 tiene un dato
- 4) F1) - Envía los datos relacionados con T_i
 A8) - Recibe los datos relacionados con T_i
 88) - Pregunta si el A8 tiene un dato
- 5) F1) - Avisa que tiene datos de fusificación para P_i
 A8) - Inicia el proceso de defusificación para T_i, P_i
 88) - Pregunta si el A8 tiene un dato
- 6) F1) - Avisa que tiene datos de fusificación para P_i
 A8) - Avisa que tiene un dato defusificado
 88) - Identifica que ya hay un dato en la entrada
- 7) F1) - Avisa que tiene datos de fusificación para P_i
 A8) - Transmite los datos defusificados para T_i
 88) - Recibe los datos relacionados con T_i
- 8) F1) - Avisa que tiene datos de fusificación para P_i
 A8) - Identifica que ya hay un dato en la entrada
 88) - Realiza autosintonización y manipulación de válvula T_i
- 9) F1) - Envía los datos relacionados con P_i
 A8) - Recibe los datos relacionados con P_i
 88) - Pregunta si el A8 tiene un dato
- 10) F1) - Realiza la búsqueda de los datos para fusificar
 A8) - Inicia el proceso de defusificación para P_i, T_i
 88) -Pregunta si el A8 tiene un dato
- 11) F1) - Realiza la determinación de los mínimos
 A8) - Avisa que tiene un dato defusificado

- 88).- Identifica que hay un dato en la entrada
- 12) F1).- Avisa que tiene un dato de fusificación para T₂
AB).- Transmite los datos defusificados para P₁
88).- Recibe los datos relacionados con P₁
- 13) F1).- Avisa que tiene un dato de fusificación para T₂
AB).- Identifica que ya hay un dato en la entrada
88).- Realiza autosintonización y manipulación de válvula P₁
- 14) F1).- Envía los datos relacionados con T₂
AB).- Recibe los datos relacionados con T₂
88).- Pregunta si el AB tiene un dato listo
- 15) F1).- Avisa que tiene un dato listo
AB).- Inicia el proceso de defusificación para T₂
88).- Pregunta si el AB tiene un dato listo
- 16) F1).- Avisa que tiene un dato listo
AB).- Avisa que tiene un dato defusificado para T₂
88).- Identifica que hay un dato a la entrada
- 17) F1).- Avisa que tiene un dato listo
AB).- Envía los datos de defusificación para T₂
88).- Recibe los datos relacionados con T₂
- 18) F1).- Avisa que tiene un dato listo
AB).- Identifica que hay un dato a la entrada
88).- Realiza autosintonización y manipulación de válvula T₂
- 19) F1).- Envía los datos relacionados con P₂
AB).- Recibe los datos relacionados con P₂
88).- Pregunta si el AB tiene un dato listo
- 20) F1).- Va al inicio nuevamente.
AB).- Inicia el proceso de defusificación para P₂
88).- Pregunta si el AB tiene un dato
- 21) F1).- Inicia el paso 1
AB).- Avisa que tiene un dato defusificado P₂
88).- Identifica que hay un dato a la entrada
- 22) F1).- Está realizando el proceso 1
AB).- Envía los datos defusificados para P₂
88).- Recibe los datos relacionados con P₂
- 23) F1).- Está realizando el proceso 2.
AB).- Va al inicio
88).- Realiza autosintonización y manipula Válvula

ANEXO 2.

CARACTERISTICAS ELECTRICAS DE CADA MICROCONTROLADOR.

CARACTERISTICAS ELECTRICAS DEL HC11.

CARACTERISTICA	SIMBOLO	1.0 MHZ		2.0 MHZ		2.1 MHZ		UNIDAD
		Min	Max	Min	Max	Min	Max	
Frecuencia de operación	f_0	dc	1 0	dc	2 0	dc	2.1	MHZ
Periodo de reloj E	t_{EVE}	1000	-	500	-	476	-	nS
Frecuencia del cristal	f_{XTAL}	-	4.0	-	8.0	-	8.4	MHZ
Frecuencia del oscilador externo	$4f_0$	dc	4.0	dc	8.0	dc	8.4	MHZ
Procesador de control de tiempo	t_{LCS}	-	200	-	25	-	69	nS
Amplitud del pulso del reset de entrada	pW_{RSTL}	8	-	8	-	8	-	t_{CR}
Tiempo de entrada mínimo	t_{MPH}	1	-	1	-	1	-	nS
Modo de programación organizado de tiempo.	t_{WRS}	2	-	2	-	2	-	nS
Modo de programación de tiempo	t_{WRS}	0	-	0	-	0	-	nS
Interrupción de amplitud de pulso	pW_{WRQ}	1020	-	520	-	496	-	nS
Tiempo organizado de reconocimiento y espera.	t_{WRS}	-	4	-	4	-	4	t_{CR}
Temporizador de captura de entrada de ancho del pulso.	pW_{TIM}	1020	-	520	-	496	-	nS

CARACTERISTICAS TERMICAS

Resistencia térmica.	simbolo	valor	unidad
Paquete de plástico cuadrado de 68 pines	θ_{JA}	50	$^{\circ}\text{C/W}$

El promedio de la temperatura de la junta del chip T_J , en $^{\circ}\text{C}$ puede ser obtenida de:

$$T_J = T_A + (P_C - \theta_{JA}) \text{ Donde:}$$

T_A = Temperatura ambiente en $^{\circ}\text{C}$.

θ_{JA} = Resistencia térmica del paquete, ambiente a junta, $^{\circ}\text{C/W}$

$P_C = P_{INT} + P_{E/S}$

$P_{INT} = I_{DD} \times V_{DD}$ watts. Potencia interna del chip.

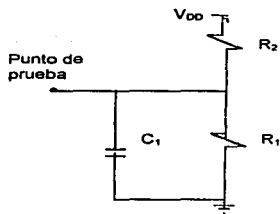
$P_{E/S}$ = Disipación de potencia sobre los pines de entrada/salida

Para algunas aplicaciones $P_{E/S} < P_{INT}$ y puede ser despreciada

ESPECIFICACIONES ELÉCTRICAS

	símbolo	valor	unidad
Fuente de voltaje	V _{DD}	-0.3 a +7.0	V
Entrada de voltaje	V _{IN}	-0.3 a +7.0	V
Rango de operación de temperatura	T _A	TL a TH	°C
Rango de almacenamiento de temperatura	T _{STG}	-40 a 85	°C
Drenaje de corriente por pin	I _{DD}	-55 a 150	mA
		25	

CARACTERÍSTICAS	SIMB.	MIN.	MAX.	UNID.
Voltaje de salida. Todas las salidas I _{cariga} = ±10 µA. Todas las salidas MODA excepto Reset	V _{OL} V _{OH}	----	0.1 V _{DD} -0.1	Volts.
Voltaje alto a la salida I _{cariga} = -0.8 mA, V _{DD} = 4.5 V	V _{OH}	V _{DD} - 0.8	-----	Volts.
Voltaje bajo de salida. Todas las salidas excepto XTAL. I _{cariga} = 1.8mA	V _{OL}	-----	0.4	Volts
Voltaje de entrada alto. Todas las entradas excepto RESET.	V _{IH}	0.7xV _{DD} 0.8xV _{DD}	V _{DD} +0.3 V _{DD} +0.3	Volts
Voltaje de entrada bajo (todas las entradas)	V _{IL}	V _{SS} - 0.3	0.2xV _{DD}	Volts
Puertos E/S tres estados PA7-PA0, PC7-PC0, PD5-PD0, PG7-PG0 V _{IN} = V _{IH} ó V _{IL} , MODALIR, RESET	I _{OZ}	-----	±10	µA
Corriente de entrada V _{IN} = V _{DD} ó V _{SS} , IRQ, XIRQ V _{IN} = V _{DD} ó V _{SS} , MODB/V _{stby}	I _{IN}	-----	±1	µA
RAM stanby, voltaje apagado	V _{BB}	-----	4.0	Volts
RAM stanby corriente apagada	I _{SB}	-----	20	µA
Alimentación total de corriente RUN.				
Modo chip sencillo	I _{DD}	-----	15	mA
Modo no multiplexado expandido		-----	27	mA



Pins	R ₁	R ₂	C ₁
4xout	3.26 kΩ	2.38 kΩ	30pF
PA7-PA3	3.26 kΩ	2.38 kΩ	90pF
PB7-PB0			
PC7-PC0			
PD5-PD0			
PF7-PF0			
PG7-PG0			
E, RAW			
PD4 - PD1	3.26 kΩ	2.38 kΩ	200pF

ESPECIFICACIONES ELECTRICAS DEL 8088

CARACTERISTICA	SIMBOLO	VALOR	UNIDAD
Fuente de voltaje	V _{DD}	- 0.3 a +7	volts
Voltaje de entrada	V _{in}	- 0.3 a +7	volts
Rango de operación de temperatura	T _A	T _L a T _H - 40 a 85	°C
Rango de almacenamiento de temperatura	T _{seg}	- 55 a 150	°C
Drenaje de corriente por pin excluyendo: V _{DD} Vas, V _{IN4} , y V _{RL}	I _o	25	mA

CARACTERISTICAS TERMICAS

CARACTERISTICA	SIMBOLO	VALOR	UNIDAD
Resistencia térmica	θ _{JA}	50	°C/W

CONSIDERACIONES DE POTENCIA

Promedio de temperatura de la juntura T_J , en grados centígrados puede ser obtenido de:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \dots\dots(1)$$

donde:

T_A = Temperatura ambiente en grados centígrados
 θ_{JA} = Resistencia térmica del paquete, juntura a ambiente, °C/W
 $P_D = P_{int} + P_{ES}$
 $P_{int} = I_{DD} \times V_{DD}$, Watts - potencia interna del chip
 P_{ES} = Disipación de potencia sobre los pines de entrada/salida, Watt
 Para más aplicaciones $P_{ES} < P_{int}$ y puede ser despreciable.

Lo siguiente es una relación aproximada entre P_D y T_J (si P_{ES} es despreciable)

$$P_D = k + (T_J + 273 \text{ °C}) \dots\dots(2)$$

Resolviendo ecuaciones (1) y (2) para k nos da

$$k = P_D(T_A + 273) + \theta_{JA}P_D^2 \dots\dots(3)$$

Donde k es una constante perteneciente a la parte particular. k puede ser determinada por la ecuación (3) midiendo P_D (en equilibrio) para una T_A conocida.

Usando este valor de k los valores de P_D y T_J pueden ser obtenidos resolviendo las ecuaciones (1) y (2) iterativamente para cualquier valor de T_A

CARACTERISTICAS ELECTRICAS DE D.C.

$$(V_{DD} = 5V_{DC} \pm 10\%, V_{SS} = 0V_{DC}, T_A = T_L \text{ a } T_H)$$

Características	simbolo	Min	Max	unidad
Voltaje de salida todas las salidas $I_{LOAD} = \pm 10 \mu A$	V_{OL}	-----	0.1	volts
todas las salidas excepto RESET y MODA.	V_{OH}	$V_{DD} - 0.1$	-----	volts
Características	simbolo	Min	Max	Unidad
Alto voltaje de salida todas las salidas $I_{load} = -0.8 \text{ mA}$, $V_{DD} = 4.5$ excepto RESET XTAL Y MODA	V_{OH}	$V_{DD} - 0.8$	-----	Volts
Voltaje bajo de salida todas las salidas $I_{LOAD} = 1.6 \text{ mA}$ excepto XTAL	V_{OL}	-----	0.4	Volts
Voltaje alto de entrada. Todas las entradas excepto RESET	V_{IH}	$0.7 \times V_{DD}$ $0.8 \times V_{DD}$	V_{DD} V_{DD}	Volts
Voltaje bajo de entrada. Todas las entradas	V_{IL}	V_{SS}	$0.2 \times V_{DD}$	Volts
Puertos E/S 3 estados. PA3, PA7, PC0 - PC7 PD0 - PD5, AS/STRA, MODA/LIR, RESET	I_{OZ}	-----	± 10	μA
Corriente de entrada	I_{in}	-----	± 1 ± 10	μA
$V_{in} = V_{DD}$ ó V_{SS} PA0 - PA3, IRQ X IRQ $V_{in} = V_{DD}$ ó V_{SS} MODB/VSTBY				

BIBLIOGRAFIA.

- [1] FUZZY SYSTEMS THEORY AND ITS APPLICATIONS, Toshiro Terano, Kiyoji Asai, Michio Sugeno, Academic Press, Inc., San Diego California, 1987.
- [2] FUZZY LOGIC AND CONTROL, Mohammad Jamshidi, Nader Vadiée, Timothy J. Ross, Prentice-Hall, Inc., Englewood Cliffs N J, USA, 1993.
- [3] FUZZY SETS AND APPLICATIONS, Selected Papers by L.A.Zadeh, Edited by R.R Yager, S.Ovchinnikov, R.M.Tong, H.T.Nguyen, John Wiley & Sons, USA, 1987.
- [4] A MICROPROCESSOR BASED FUZZY CONTROLLER FOR INDUSTRIAL PURPOSES, T.Yamazaki and M.Sugeno, Industrial Applications of Fuzzy Control, Edited by Michio Sugeno, Elsevier Science Publishing Company Inc., New York, N. Y., USA, 1992. ,
- [5] FUZZY LOGIC, Bart Kosko and Satoru Isaka, Scientific American, 1993.
- [6] FUZZY FUNDAMENTALS, Earl Cox, Spectrum, 1992. ,
- [7] FUZZY LOGIC IN CONTROL SYSTEMS:FUZZY LOGIC CONTROLLER-PARTS 1 AND 2, Chuen Chien Lee, University of California, Berkeley, California, IEEE, 1990.
- [8] THEORY OF THE FUZZY CONTROLLER: AN INTRODUCTION, James J. Buckley, Elsevier Science Publishers, 1992.
- [9] INTRODUCTION TO THE THEORY OF FUZZY SUBSETS VOLUME1, A. Kaufmann, Academic Press, New York, N.Y., 1975.
- [10] FUZZY CONTROL AND FUZZY SYSTEMS, Witold Pedrycz, John Wiley & Sons, Inc, Winnipeg Canada, 1989.
- [11] THE INTEL MICROPROCESSORS 8086/8088, 80186, 80286, 80386, 80486, ARCHITECTURE, PROGRAMMING AND INTERFACING, Brey B. Bary, Segunda edición, Merril, 1991.
- [12] M68HC11 REFERENCE MANUAL, Motorola Inc., 1991.
- [13] LINEAR DATABOOK NATIONAL SEMICONDUCTOR, USA, 1 980.
- [14] THE TTL DATA BOOK FOR DESIGN ENGINEERS, Texas Instruments Incorporated, USA, 1981.

- [15] **NEURONAL NETWORKS AND FUZZY SYSTEMS**, Bart Kosko, Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [16] **FUZZY SYSTEMS AN OVERVIEW**, Toshinori Munakata, Yashuan jani, Communications of the ACM, 1994.
- [17] **STABILITY ANALYSIS AND DESING OF FUZZY CONTROL SYSTEMS**, Kasuo Tanaka and Michio Sugeno, Elsevier Science Publishers, 1992.
- [18] **A STABILITY APPROACH TO FUZZY CONTROL DESING FOR NONLINEAR SYSTEMS**, Guang-Chyan Hwang and Shih-Chang Lin, Elsevier Science Publishers, 1990.
- [19] **MICROCONTROLLER TECHNOLOGY THE 68HC11**, Peter Spasov, Prentice - Hall, 1993.
- [20] **ARQUITECTURA DE COMPUTADORAS Y PROCESAMIENTO PARALELO**, Kai Hwang, Faye A. Briggs, Mc Graw - Hill, 1988.
- [21] **TRANSDUCERS, THEORY AND APPLICATIONS**, John A. Alloca, Allen Stuart, Resson Publishing company, A Prentice - Hall company, Restonvirginia, 1984
- [22] **DISEÑO Y DESARROLLO DE UN SISTEMA PARA EL CONTROL DISTRIBUIDO DE PROCESOS USANDO UN SISTEMA OPERATIVO EN TIEMPO REAL**, Tesis de Licenciatura, Facultad de Ingeniería, UNAM, México, D.F., 1995.
- [23] **DISEÑO DE UN SISTEMA DE MONITOREO Y CONTROL DISTRIBUIDO EN TIEMPO REAL**, Tesis de Maestría, Facultad de Ingeniería, UNAM, México, D.F., 1996.
- [24] **THE 8088 AND 8086 MICROPROCESORS, PROGRAMMING, INTERFACING, SOFTWARE, HARDWARE AND APPLICATIONS**, Walter A. Triebel, Avtar Singh, Prentice - Hall, 1991.