

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

CONTROL DIFUSO EN TIEMPO REAL DE UN NIVEL DE LÍQUIDO

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRONICA Y CONTROL**

NELSON FERNANDO FALCONI TELLO

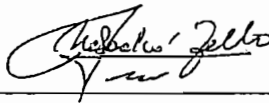
DIRECTOR: Prof. MSc. MARCO BARRAGÁN B.

Quito, Noviembre 2000

DECLARACIÓN

Yo , Nelson Fernando Falconí Tello, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

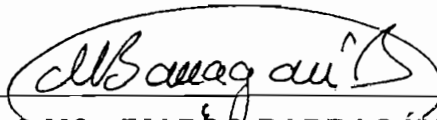
A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



**NELSON FERNANDO
FALCONI TELLO**

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Nelson Fernando Falconí Tello, bajo mi supervisión.

A handwritten signature in black ink, enclosed within a hand-drawn oval. The signature is cursive and appears to read 'M. Barragán B.'.

Prof. MSc. MARCO BARRAGÁN B.

DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Agradezco al Ing. Marco Barragán por su paciencia, y su ayuda para la realización de este proyecto con su acertada dirección y consejos.

A familiares y amigos que han contribuido con su granito de arena para la culminación de este trabajo.

DEDICATORIA

A Dios por ser él quién me ha iluminado. A mis padres por ser ellos los que me han apoyado incondicionalmente, que se han sacrificado para verme terminar mi carrera, lo que soy ahora se los debo a ellos.

CONTENIDO

CAPITULO 1

TEORIA DIFUSA

1.1 HISTORIA DE LA LÓGICA DIFUSA	1
1.2 CONJUNTOS DIFUSOS	4
1.2.1 UNIVERSO	6
1.2.2 FUNCIONES DE PERTENENCIA	6
1.2.3 VARIABLES LINGÜÍSTICAS	10
1.3 OPERACIONES ENTRE CONJUNTOS DIFUSOS	12
1.3.1 MODIFICADORES	15
1.3.2 RELACIONES ENTRE CONJUNTOS	15
1.4 LÓGICA DIFUSA	20
1.4.1 CONECTIVOS	20
1.4.2 IMPLICACION Y REGLA DE INFERENCIA	21

CAPITULO 2

CONTROL DIFUSO

2.1 OBTENCIÓN DE LA BASE DE CONOCIMIENTO	28
2.2 ESTRUCTURA DE UN CONTROLADOR DIFUSO	31
2.2.1 PRE-PROCESAMIENTO	31
2.2.2 FUSIFICACIÓN	32
2.2.3 BASE DE CONOCIMIENTO	32
2.2.4 MECANISMO DE INFERENCIA	35
2.2.5 DEFUSIFICACION	37
2.2.6 POST-PROCESAMIENTO	38

CAPITULO 3

DISEÑO DEL SOFTWARE

3.1 DISEÑO DEL HARDWARE	40
3.1.1 SENSORES DE NIVEL DE LIQUIDO	40
3.1.2 CARACTERISTICAS DEL SENSOR Y ACONDICIONAMIENTO	45
3.1.3 TARJETA DE ADQUISICIÓN DE DATOS	46
3.1.4 ACONDICIONAMIENTO DE LA SEÑAL	49
3.1.5 FUENTES DE ENERGIA UTILIZADAS	52
3.2 DISEÑO DEL SOFTWARE	54
3.2.1 ACERCA DE LOS PROGRAMAS UTILIZADOS	54
3.2.2 CONSTRUCCION DE LIBRERIAS DINAMICAS	54
3.2.3 PASOS PARA LA IMPLEMENTACION	55
3.2.4 INTERFACES GRAFICAS Y DIAGRAMAS DE FLUJO	63

CAPITULO 4

RESULTADOS EXPERIMENTALES

4.1 CONJUNTO DE PRUEBAS I	71
4.2 CONJUNTO DE PRUEBAS II	74
4.3 CONJUNTO DE PRUEBAS III	77
4.4 CONJUNTO DE PRUEBAS IV	80
4.5 CONJUNTO DE PRUEBAS V	83
4.6 CONJUNTO DE PRUEBAS VI	86
4.7 CONJUNTO DE PRUEBAS VII	89
4.8 CONJUNTO DE PRUEBAS VIII	92

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES	95
5.2 RECOMENDACIONES	97

REFERENCIAS BIBLIOGRAFICAS	99
----------------------------	----

ANEXOS

ANEXO A. MANUAL DE USUARIO	101
ANAEXO B. LISTADO DE PROGRAMAS	106

RESUMEN

En este trabajo se ha realizado el control difuso en tiempo real de un nivel de líquido, en el que se ha obtenido resultados satisfactorios, que muestran la utilidad de este tipo de tecnología.

Los capítulos están redactados, en el orden en el que se realizó este proyecto. Así en lo que tiene que ver con la investigación y el estudio del tema están: el capítulo 1 que trata acerca de la teoría de la lógica difusa, el capítulo 2 acerca de las partes que conforman un controlador difuso. En lo relacionado con el diseño e implementación del control se halla en el capítulo 3; en el capítulo 4 se muestran los resultados que se ha obtenido; y por último el capítulo 5 en el que se presentan las conclusiones y recomendaciones.

PRESENTACIÓN

Las tecnología de vanguardia como la inteligencia artificial , o los sistemas expertos son cada vez mejorados para realizar controles óptimos en cualquier tipo de sistemas.

Estas técnicas almacenan su conocimiento en forma de reglas de inferencia basadas en el conocimiento del experto, de aquí que la lógica difusa (fuzzy) es también un tipo de sistema experto y una de las técnicas de inteligencia artificial.

Es por este motivo, que se justifica la realización del presente proyecto de titulación, con el fin de investigar estas nuevas tecnologías, que tienen un futuro promisorio no solo en la electrónica (robótica), sino también en todos los ámbitos del que hacer humano.

CAPITULO 1

TEORIA DIFUSA

1.1 HISTORIA DE LA LOGICA DIFUSA

La lógica difusa es una ciencia relativamente nueva, aunque la idea de difuso o vago ya había sido discutida. Filósofos de la antigüedad como Aristóteles creador de la lógica formal, admitía la existencia de diferentes grados de verdad y falsedad.

Desde el siglo XVIII pensadores de la talla de Berkeley, Hume, Kant, Bayes y otros volcaron sus ideas para discutir la lógica difusa.

Los datos de orígenes de la lógica difusa se ven descubiertas en los años veinte, cuando Bertrand Russell descubrió una antigua paradoja griega que dice:

Un cretano afirma que todos los cretanos mienten. ¿ Si es así, el está mintiendo?. Si él miente, entonces él esta diciendo la verdad y no miente. Si él no miente, entonces él dice la verdad y, además , él miente.

En cualquier caso que todos los cretanos mienten o que todos los cretanos no mienten_ existe una contradicción, porque ambas frases son verdaderas y falsas. Russell encontró que esta misma paradoja aplicada a la teoría de conjuntos debería ser totalmente falsa o totalmente verdadera .

La lógica difusa superó este problema de la lógica clásica , pues declaraciones de este tipo pueden ser interpretadas como verdaderas y falsas a la vez. Por lo tanto aplicando la lógica difusa a la paradoja griega tendrían estas declaraciones su

grado de verdad y falsedad; los cretanos dicen la verdad el 50% del tiempo, y mienten el 50% del tiempo, y está sería una interpretación desde el punto de vista de lógica difusa.

Alrededor de 1920s, independientemente de Bertrand Russel , un matemático polaco llamado Jan Lukasiewicz comenzó trabajando con lógica de múltiples valores, por lo cual creo una fracción binaria de valores comprendidos entre 1 lógico y 0 lógico. En 1937 el filósofo Max Black, en un artículo aplicó esta lógica de multivalor a listas (o conjuntos) y dibujo curvas difusas, llamándoles conjuntos difusos.

Posteriormente, el Dr. Lofti Zadeh, profesor del Departamento de Ingeniería Eléctrica de la Universidad de California en Berkeley, 1965 propuso un método de razonamiento abstracto similar al patrón de pensamiento humano para representar los problemas del mundo real, y con este fin creo la lógica difusa.

A comienzos de los 70s, el ingeniero británico Ebrahim Mandani, basado en la teoría de Zadeh, desarrolla el primer sistema de control difuso práctico, aplicado a una máquina de vapor.

El sistema de Mandani combinaba la experiencia de un operador humano con un conjunto de reglas lógicas para controlar automáticamente la cantidad de vapor y la temperatura de la caldera de acuerdo a la velocidad y presión de la máquina. Las dos entradas (velocidad y presión) eran procesadas de acuerdo a un cierto algoritmo y producían dos salidas (cantidad de vapor y temperatura) que actuaban sobre el proceso en la forma deseada. A finales de los 70s, los ingenieros daneses Lauritz Peter Holmblad y Jens-Jurgen Ostergaard desarrollan el primer sistema de control difuso comercial, destinado a una planta de cemento.

A pesar de que han transcurrido cerca de tres décadas desde su creaciones recientemente que el mundo occidental esta reconociendo el valor de la lógica difusa; uno de los motivos que explican esta situación es que la misma palabra difuso es algo confuso, razón por la cual alejaba a los científicos a utilizarla, y

además por hecho de que no necesitaba de modelos matemáticos como lo hacía el control clásico. En contraste con este punto de vista, en Japón ocurrió todo lo contrario, en este país evolucionó más rápidamente en la fase experimental y es así que en la actualidad Japón es el líder a nivel mundial en la producción de aplicaciones basadas con lógica difusa.

1.2 CONJUNTOS DIFUSOS

Los conjuntos difusos son un desarrollo adicional del concepto matemático de un conjunto, y así es conveniente recordar primero el concepto formal de conjunto.

CONJUNTO.- Un conjunto es una colección de objetos que pueden ser tratados como un todo. Sea un conjunto A en un universo de discurso U , puede ser definido de las siguientes formas:

- Como una lista de todos sus miembros.
- Por una propiedad de sus miembros.

$A = \{ x / x \text{ tenga alguna condición} \}$

- Empleando la función de membresía o pertenencia cero-uno para A , denotada $\mu_A(x)$, tal que :

$$\mu_A(x) = 1 \text{ si } x \in A \text{ y:}$$

$$\mu_A(x) = 0 \text{ si } x \notin A.$$

Pero el conjunto " gente joven", ya es algo vago .Así un bebe de un año de edad será claramente un elemento de este conjunto, mientras que una persona que tenga 100 años no será elemento de éste conjunto, pero que sucede con la gente de edad de 20,30, o 40 años?

Zadeh propuso que los diferentes valores que pueda adoptar una variable del mundo real se subdividen en grupos y a cada valor dentro de un grupo se le asigne una cuota de pertenencia al mismo denominada *grado de pertenencia* o membresía .Así al grado de membresía de todos sus miembros(elementos) es lo que describe a un conjunto difuso. El grado de membresía es entonces un número real comprendido entre uno y cero, y se le nota con μ . En la figura 1.1

se indica como se podría representar el conjunto gente joven asociando diferentes grados de pertenencia , a las personas según su edad.

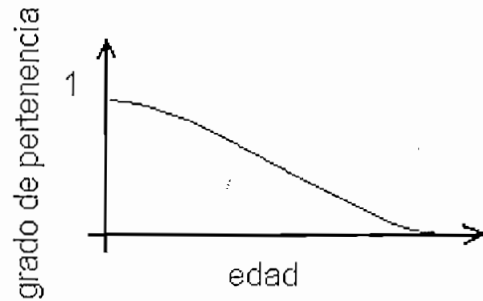


Figura1.1.Representación del conjunto edad según diferentes grados de pertenencia.

Con estas ideas, la definición formal de un conjunto difuso es la siguiente:

Un conjunto difuso F en U (universo de discurso) , puede ser representado como un conjunto de pares ordenados de un elemento genérico x de un conjunto claro, y por su grado de pertenencia μ_F de todo esto:

$$F = \{ (x, \mu_F(x)) / x \in U \} \quad \text{expresión[1.1]}$$

F es comúnmente escrito de una manera más formal como:

$$F = \int_U \mu_F(x)/x \quad \text{expresión [1.2]}$$

Si U es continuo, en esta ecuación, el signo de la integral no significa integración; sino que denota la colección de todos los puntos $x \in U$ (continuo) asociados a su grado de pertinencia $\mu_F(x)$, o

$$F = \int_U \mu_F(x) / x. \quad \text{expresión [1.3]}$$

Si U es discreto al igual que en el caso anterior el signo de sumatoria no denota la adición aritmética; lo que da a entender es la colección de todos los puntos $x \in U$ relacionados con su grado de pertenencia $\mu_F(x)$, diferente de cero.

1.2.1 UNIVERSO

Los elementos de un conjunto difuso, como ya se mencionó, son tomados de un universo de discurso o simplemente universo.

El universo contiene todos los elementos que serán considerados.

Una aplicación del universo es la siguiente: si se toman medidas de voltaje en el cual valores negativos no tienen significado, estos datos son errados y por lo tanto no entran a formar parte del universo del discurso.

El universo no solo pueden ser de valores numéricos, sino también expresiones no numéricas, así por ejemplo el gusto, no podría ser medido en una escala numérica, o como otros ejemplos que son tomados de palabras expresadas por el o los individuos (variables lingüísticas), como {amargo, dulce, salado,etc}.

1.2.2 FUNCIONES DE PERTENENCIA

Cada elemento en el universo del discurso es miembro de un conjunto difuso con algún grado de pertenencia, como ya se mencionó a este grado de pertenencia $F(x)$ se le denomina función de membresía o pertenencia.

Las funciones de pertenencia más comúnmente usadas son triangulares, trapezoidales, gaussianas, singleton[9].

A continuación se definirá solamente a las funciones de pertenencia más utilizadas:

Función Trapezoidal : La función de membresía trapezoidal, figura 1.2 es descrita como:

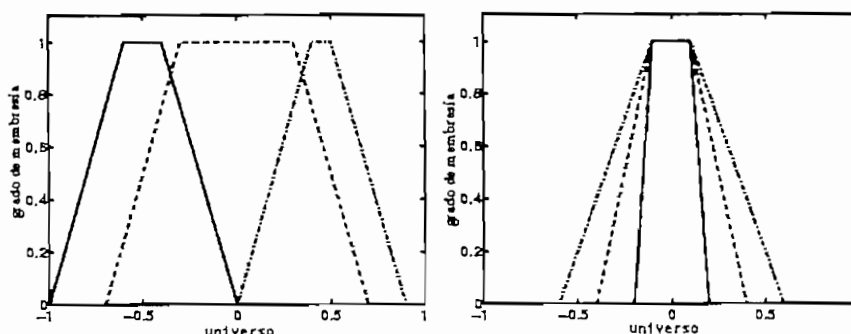


Figura 1.2. Funciones de membresía trapezoidales.

$$f(x) = \begin{cases} 0 & , x \leq x_l^- \\ (x - x_l^-) / (x_u^- - x_l^-) & , x_l^- < x \leq x_u^- \\ 1 & , x_u^- \leq x < x_u^+ \\ (x_u^+ - x) / (x_l^+ - x_u^+) & , x_u^+ \leq x < x_l^+ \\ 0 & , x \geq x_l^+ \end{cases}$$

expresiones [1.4]

Función Triangular : La función de membresía triangular , figura 1.3 es descrita como:

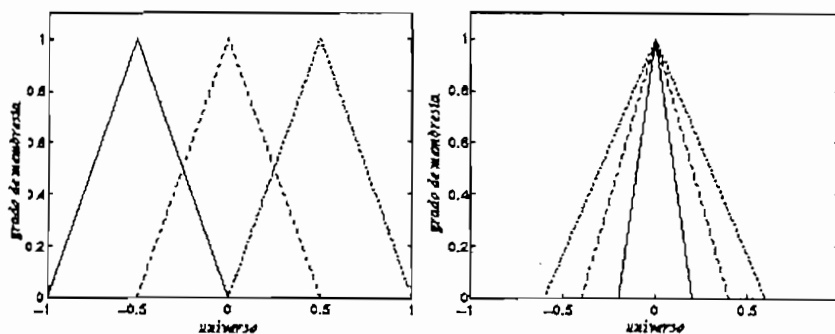


Figura 1.3. Funciones de membresía triangulares.

$$f(x) = \begin{cases} 0 & , x \leq x_l^- \\ (x - x_l^-) / (x_u - x_l^-) & , x_l^- < x \leq x_u \\ (x_u - x) / (x_l^+ - x_u) & , x_u \leq x < x_l^+ \\ 0 & , x \geq x_l^+ \end{cases}$$

expresiones [1.5]

Notemos que la función de membresía triangular es un caso especial de la trapezoidal, cuando $x_u^- = x_u^+$ en la expresión [1.4]

Función Gaussiana :La función de membresía gaussiana figura 1.4 es descrita por:

$$f(x) = e^{-((x-\bar{x})/\sigma)^2} \quad \text{expresión [1.6]}$$

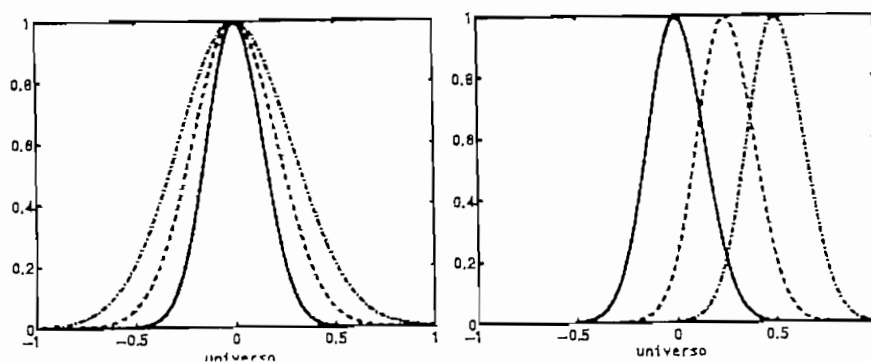


Figura 1.4. Funciones de membresía Gaussianas.

Aquí se puede apreciar que \bar{x} determina el centro de la función de pertenencia, y mientras que σ controla en cierta forma el ancho .

Función Sigmoideal : Una función de membresía sigmoide, figura 1.5 es descrita por:

$$f(x) = 1 / (1 + e^{-\alpha(x-\bar{x})}) \quad \text{expresión [1.7]}$$

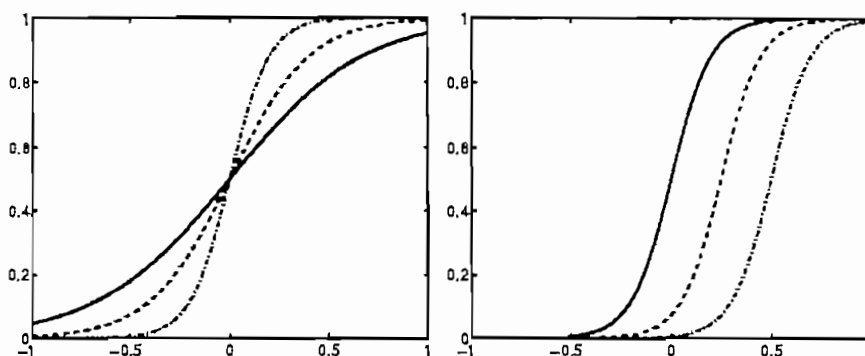


Figura 1.5. Funciones de membresía sigmoideas.

Función Singleton : La función de membresía singleton ,figura 1.6 es descrita por:

$$f(x) = \begin{cases} 1 & , x = \bar{x} \\ 0 & , x \neq \bar{x} \end{cases}$$

expresiones [1.8]

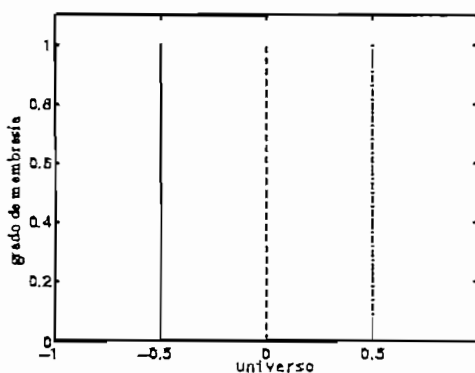


Figura 1.6. Funciones de membresía singleton.

1.2.3 VARIABLES LINGÜÍSTICAS

Así expresiones como una variable algebraica toma números como valores, una variable lingüística toma palabras u oraciones como valores. El conjunto de valores que esta puede tomar se llama conjunto de términos. Una variable lingüística esta caracterizada por $(x, T(x), U, G)$ [1].

Donde :

x es el nombre de la variable.

$T(x)$ es el conjunto de valores lingüísticos que la variable puede tomar (etiquetas).

U es el universo de discurso para la variable .

G es la regla semántica asignada a un conjunto difuso en U para cada valor lingüístico en T(x).

Para aclarar esta definición se la ilustra con el siguiente ejemplo:

Si la velocidad de un carro es interpretada como una variable lingüística, entonces:

x "es velocidad".

T(x) o lo que es lo mismo T(velocidad) puede tomar los valores lingüísticos = {lento, medio, rápido}

U La velocidad del carro es limitada a [0,100] kmph.

G El significado semántico de lento, medio, rápido son dados por los conjuntos difuso $\mu_{\text{lento(velocidad)}}$, $\mu_{\text{medio(velocidad)}}$, $\mu_{\text{rápido(velocidad)}}$.

Es decir se asignarán las etiquetas lento, medio, rápido a las funciones de membresía $\mu_{\text{lento}(v)}$, $\mu_{\text{medio}(v)}$, $\mu_{\text{rápido}(v)}$ respectivamente, la figura 1.7 muestra lo anteriormente dicho.

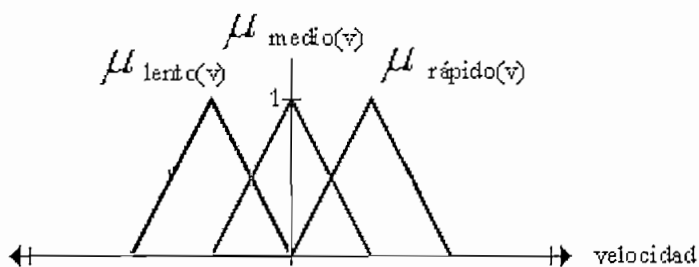


Figura 1.7. Funciones de pertenencia con etiquetas

1.3 OPERACIONES ENTRE CONJUNTOS DIFUSOS

En la teoría de conjuntos se tiene el siguiente grupo de operaciones básicas:

La *intersección* del conjunto A y B, es definido como:

$$A \cap B = \{x \in U / x \in A \text{ y } x \in B\} \quad \text{expresión [1.9]}$$

La *unión* de A y B es definido como:

$$A \cup B = \{x \in U / x \in A \text{ o } x \in B \text{ o ambos}\} \quad \text{expresión [1.10]}$$

El *complemento* de A, es definido como:

$$\bar{A} = \{x \in U / x \notin A\}. \quad \text{expresión [1.11]}$$

Además las dos fundamentales leyes aristotélicas para la teoría de conjuntos son:

1.-*La Ley de Contradicción:*

$$A \cup \bar{A} = U \quad \text{expresión [1.12]}$$

2.-*Ley de Exclusión :*

$$A \cap \bar{A} = \emptyset \quad \text{expresión [1.13]}$$

Desde esta perspectiva se pasa a ver el grupo de operaciones en conjuntos difusos ; las funciones de membresía son componentes cruciales de un conjunto difuso, ya que una función de membresía lo caracteriza . Sería entonces natural definir operaciones sobre conjuntos difusos por medio de sus funciones de membresía.

A continuación algunas definiciones importantes:

t-norma .- Denotada con * es una operación de intersección (mínimo o producto) en conjuntos difusos[10].

s-norma .- Denotada con $+$, es una operación de unión en conjuntos difusos[10]. Otras formas de representar las normas s y t se hallan en [2] y [10].

La Unión de Conjuntos difusos. Sean A y B dos conjuntos difusos, la unión de A y B será otro conjunto difuso C y estará dado por:

$$C = A \cup B \text{ y:}$$

$$\mu_C(x) = \text{máximo} [\mu_A(x), \mu_B(x)] \text{ o lo que es lo mismo:}$$

$$\mu_{A \cup B}(x) = \text{máximo} [\mu_A(x), \mu_B(x)] \text{ expresión [1.14]}$$

A este tipo de operación también se le conoce como conorma t o norma s. Para entender esta operación se realizara el siguiente ejemplo:

$$\text{Sea } A = \{ (0, 0.5), (1, 0.3), (2, 0.5), (4, 1) \} \text{ y}$$

$$B = \{ (0, 0.7), (1, 0.5), (2, 0.9), (4, 0.2) \}.$$

Entonces :

$$C = A \cup B = \{ \max(0.5, 0.7)/0, \max(0.3, 0.5)/1, \max(0.5, 0.9)/2, \max(1.0, 0.2)/4 \}$$

$$C = \{ 0.7/0, 0.5/1, 0.9/2, 1/4 \}$$

$$C = \{ (0, 0.7), (1, 0.5), (2, 0.9), (4, 1.0) \}.$$

La Intersección de Conjuntos difusos.- Sean A y B dos conjuntos difusos, la intersección de A y B será otro conjunto difuso C y estará definido por:

$$C = A \cap B \text{ de donde:}$$

$$\mu_C(x) = \text{mínimo} [\mu_A(x), \mu_B(x)], x \in U \text{ o lo que es lo mismo}$$

$$\mu_{A \cap B}(x) = \text{mínimo} [\mu_A(x), \mu_B(x)] \text{ expresión [1.15]}$$

A esta clase de operación se le conoce como norma $-t$. A continuación un ejemplo para comprender esta operación.

Sean A y B del ejemplo anterior entonces:

$$C = A \text{ I } B = \{ \min (0.5, 0.7) / 0, \min (0.3, 0.5) / 1, \min (0.5, 0.9) / 2, \min (1.0, 0.2) / 4 \}$$

$$C = \{ 0.5 / 0, 0.3 / 1, 0.5 / 2, 0.2 / 4 \}$$

$$C = \{ (0, 0.5), (1, 0.3), (2, 0.5), (4, 0.2) \}.$$

pero también existe otra variante que es la intersección por producto:

$$\mu_{A \text{ I } B}(x) = [\mu_{A(x)} \cdot \mu_{B(x)}] \text{ donde } x \in U ; \text{ expresión [1.16]}$$

aplicada al ejemplo de la intersección quedaría como :

$$C = \{ (0, 0.35), (1, 0.15), (2, 0.45), (4, 0.2) \}.$$

El Complemento difuso.- Tiene la siguiente forma:

$$\mu_{\bar{A}}(x) = 1 - \mu_{A(x)}. \text{ } x \in U ; \text{ expresión [1.17]}$$

Para entender esta operación se pone en consideración el siguiente ejemplo:

Sea $A = \{ (1, 0.4), (2, 0.6), (3, 0.8), (4, 1) \}$ De aquí :

$\mu_{\bar{A}}(x) = 1 - \mu_{A(x)}$. Esta expresión aplicada en A quedaría como:

$$\bar{A} = \{ (1, 1-0.4), (2, 1-0.6), (3, 1-0.8), (4, 1-1) \}$$

$$\bar{A} = \{ (1, 0.6), (2, 0.4), (3, 0.2), (4, 0) \}$$

De los ejemplos anteriores se habrá notado que , al realizar las operaciones de unión, intersección, y complemento, las funciones de membresía se ven alteradas dependiendo del operador que se utilice.

Es de mencionar también que en la lógica difusa no se cumplen las dos leyes

fundamentales de la teoría normal de conjuntos ,ya que un elemento debe estar en un conjunto o en su complemento pero no en ambos ,como sucede en conjuntos difusos.

1.3.1 MODIFICADORES

Un modificador lingüístico , es una operación que modifica el significado de un término. Por Ejemplo, en la expresión "muy cerca de 0" , la palabra **muy** modifica a la expresión "cerca de 0" que es un conjunto difuso.

Un modificador es una operación sobre un conjunto difuso. Ejemplos de otros modificadores son : pequeño , mas o menos, muy , etc. Los modificadores pueden llegar a tener efectos opuestos como lo son : más o menos.

1.3.2 RELACIONES ENTRE CONJUNTOS

Otra de las operaciones importantes en la teoría de conjuntos es el producto cartesiano que se define como:

$$A_1 \times A_2 \dots \times A_n = \{ (x_1, \dots, x_n) / x_n \in A_n \} \text{ expresión [1.18]}$$

Donde A_n son conjuntos contenidos en el universo U_n .Su operación genera un vector ordenado en el cual cada elemento x_n pertenece a A_n .

En el caso particular si $n= 2$,el producto cartesiano conduciría a la formación de pares ordenados.

Al utilizar funciones de membresía para la operación producto cartesiano quedaría de la siguiente forma [10]:

$$\mu_{A_1 \times A_2 \times \dots \times A_n} (x_1 , \dots , x_n) = \text{mínimo} \{ \mu_{A_1}(x_1), \dots , \mu_{A_n}(x_n) \}$$

o también:

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(x_1, \dots, x_n) = \mu_{A_1}(x_1) \dots \mu_{A_n}(x_n)$$

Expresiones [1.19] y [1.20]

El concepto de relación puede ser visto como una generalización de funciones, una función describe matemáticamente como dos o más elementos están relacionados, por ejemplo, $f(x) = x$, describe la dependencia de $f(x)$ en x . La función $f(x)$ puede ser evaluada para todos los valores de x .

Pero no todos los elementos pueden ser llevados a una expresión matemática, por ejemplo, un conjunto de ciudades y un conjunto de países, una forma para relacionarlos será aplicando el producto cartesiano a estos dos conjuntos para obtener pares ordenados, relacionando elementos del primer conjunto con los elementos del segundo conjunto. Es por esto que se dice que el producto cartesiano es una operación de relación[8].

Considérese el siguiente ejemplo: Sea el conjunto A formado por los siguientes deportes $A = \{\text{fútbol, tenis, natación, voley}\}$, y el conjunto B por los siguientes elementos $B = \{\text{pelota, pantaloneta, camiseta}\}$. El producto cartesiano $A \times B$ generara todas las posibles combinaciones, así:

$$A \times B = \{(\text{fútbol, pelota}), (\text{fútbol, pantaloneta}), (\text{fútbol, camiseta}), (\text{tenis, pelota}), (\text{tenis, pantaloneta}), (\text{tenis, camiseta}), (\text{voley, pelota}), (\text{voley, pantaloneta}), (\text{voley, camiseta})\}$$

Una relación R de A a B puede ser definida seleccionando pares ordenados del conjunto $A \times B$ que satisfagan una condición particular. Por ejemplo:

$$R = \{(\text{fútbol, pelota}), (\text{tenis, pantaloneta}), (\text{voley, pelota})\}$$

Relaciones difusas.-Las relaciones difusas representan el grado de presencia o ausencia de asociación, interacción entre los elementos de dos o más conjuntos difusos. Sean U y V universos de discurso, una relación difusa $R(U, V)$ es un conjunto difuso en el espacio $U \times V$, y es caracterizado por la función de membresía $\mu_R(x, y)$ donde $x \in U$ y $y \in V$, de aquí [9]:

$$R(U, V) = \{(x, y), \mu_R(x, y) \mid (x, y) \in U \times V\} \text{ expresión [1.21].}$$

La diferencia entre una relación y una relación difusa radica en el hecho de que, para la relación difusa, $\mu_R(x, y) \in [0, 1]$; mientras que $\mu_R(x, y) = 0$ o 1 para la relación [9].

Para aclarar este concepto citaremos el siguiente ejemplo:

Asúmase que los sobrinos de José, el uno llamado Juan se parece a Carlos en un grado de 0.8, y Juan se parece a Fernando en un grado de 0.9, esto es obviamente una relación que se podría representar en forma de matriz así:

$$R_1 = \begin{matrix} & \begin{matrix} \text{Carlos} & \text{Fernando} \end{matrix} \\ \text{Juan} [& \begin{matrix} 0.8 & 0.9 \end{matrix} \end{matrix}$$

Composición.- Las operaciones que se llevan a cabo entre relaciones son conocidas como operaciones de composición [9].

Del ejemplo anterior, se podría también formar una relación R_2 entre Carlos y Fernando con su tío José de la siguiente manera:

$$R_2 = \begin{matrix} & \text{José} \\ \begin{matrix} \text{Carlos} \\ \text{Fernando} \end{matrix} & \begin{pmatrix} 0.5 \\ 0.6 \end{pmatrix} \end{matrix}$$

Ahora bien, se tratara de encontrar en que grado se parece Juan a su tío José para esto, se combina la información de estas dos matrices R_1 y R_2 , con lo que se llega a:

- (i) Juan se parece (0.8) a Carlos (0.8 es el grado de pertenencia), y Carlos se parece (0.5) a José, o
- (ii) Juan se parece (0.9) a Fernando, y Fernando se parece (0.6) a José.

De (i), contiene una cadena de declaraciones, y por lo visto es razonable combinarlas con una operación de intersección, igualmente con la expresión(ii),concluyendo:

- (iii) Juan se parece en (0.5) a José, o
- (iv) Juan se parece en (0.6) a José.

Ambas, (iii) y (iv) son expresiones validas, y sería natural aplicar la operación unión, llegando a un resultado final:

- (v) Juan se parece en un grado de 0.6 a su tío José.

De este ejemplo se puede concluir que la intersección es una operación de composición.

Desde el punto de vista de la formalidades matemáticas se define a la composición entre relaciones difusas como [10]:

Sean R y S dos relaciones difusas en $U \times V$ y $V \times W$ respectivamente, la composición de $R(U, V)$ y $S(V, W)$ es una relación difusa simbolizada como $R \circ S$, y con función de membresía[10]:

$$\mu_{R \circ S}(x, z) = \sup_{y \in V} [\mu_R(x, y) * \mu_S(y, z)] \quad \text{expresión [1.22]}$$

Donde el asterisco representa una norma t (mínimo o producto), esta operación de composición se le denomina supremo asterisco; una simplificación a esta expresión se produce cuando, la relación R esta conformada por un solo conjunto difuso, en consecuencia $\mu_R(x,y)$ pasa a ser solamente $\mu_{R(x)}$, en otras palabras $V = U$, por lo que la ecuación anterior se reduce a [10]:

$$\mu_{R \circ S}(z) = \sup_{x \in U} [\mu_R(x) * \mu_S(x,z)] \quad \text{expresión [1.23]}$$

1.4 LOGICA DIFUSA

La lógica tradicional comenzó como el estudio del lenguaje tanto en argumentos, como proposiciones (una expresión que se le puede asignar un valor de verdad o falsedad).

En contraposición a la lógica tradicional, para la lógica difusa una proposición puede tomar valores verdaderos o falsos como también valores intermedios entre verdad y falsedad [9].

1.4.1 CONECTIVOS

En la conversación diaria, las expresiones(proposiciones) son conectadas con palabras tales como y, o, si-entonces, a estas palabras se les denomina conectivos.

Estos conectivos ,tienen la función de cambiar el valor de verdad de las proposiciones compuestas (unión de proposiciones simples),es por esto que estos conectivos son operaciones lógicas.

Entre los conectivos u operaciones más importantes están:

Conjunción	\wedge	“y”
Disjunción	\vee	“o”
Negación	\neg	“no”
Implicación	\Rightarrow	“si- entonces”

Considérese la siguiente expresión :Si Barcelona o Emelec pierden y Olmedo gana ,entonces el Aucas quedara fuera del primer lugar ,y yo perderé la apuesta.

La expresión compuesta es un condicional de la forma $r \Rightarrow s$. El antecedente esta compuesto de tres proposiciones p (“Barcelona pierde”),c(“ Emelec pierde”),y g(“ Olmedo gana”).El consecuente es la conjunción de d (“Aucas estará fuera del

primer lugar”) y b (“Yo perderé la apuesta”). Así la original expresión puede ser simbolizada por:

$$((p \vee c) \wedge g) \Rightarrow (d \wedge b).$$

1.4.2 IMPLICACION Y REGLA DE INFERENCIA

A continuación se citará algunos conceptos que al ser abstractos o difíciles de discernir es necesarios incluirlos ,para entender mejor el comportamiento de los controladores difusos que se verá más adelante.

En la tradicional lógica proposicional una implicación es definida como [9]:

$$p \Rightarrow q \Leftrightarrow \neg [p \wedge (\neg q)] \quad \text{expresión [1.24]}$$

pero también se puede expresar como:

$$p \Rightarrow q \Leftrightarrow (\neg p) \vee q \quad \text{expresión [1.25]}$$

La tabla de verdad se muestra a continuación

p	q	$p \Rightarrow q$
F	F	V
F	V	V
V	F	F
V	V	V

De las expresiones [1.24] y [1.25] se pueden llegar a obtener dos funciones de membresía para $\mu_{p \Rightarrow q}(x,y)$.

$$\mu_{p \Rightarrow q}(x,y) = 1 - \mu_{p \cap \bar{q}}(x,y). \quad \text{expresión [1.26]}$$

$$= 1 - \text{mínimo}[\mu_{p(x)}, 1 - \mu_{q(y)}] \quad \text{expresión [1.27]}$$

$$\mu_{p \Rightarrow q}(x,y) = \mu_{\bar{p} \cup q}(x,y).$$

$$= \text{máximo} [1 - \mu_{p(x)}, \mu_{q(y)}] \quad \text{expresión [1.28]}$$

Al utilizar el producto para la intersección en la expresión [1.27], se obtendrá:

$$\mu_{p \Rightarrow q}(x,y) = 1 - \mu_{p(x)} (1 - \mu_{q(y)}) \quad \text{expresión [1.29]}$$

Es decir que la implicación proporcionan sentencias del tipo "Si p Entonces q". Para llegar a conclusiones de una regla de conocimiento, se necesita de un mecanismo que pueda producir una salida de un conjunto de reglas del tipo "Si-Entonces", esto se logra usando la regla composicional de inferencia, (el verbo inferir significa concluir de pruebas), o tener una consecuencia lógica[9].

En la lógica tradicional proposicional se utiliza la famosa regla de inferencia el **Modus Ponens** que realiza el siguiente razonamiento[10]:

Premisa 1: "x es A"

Premisa 2: "Si x es A entonces y es B"

Consecuencia : " y es B".

Mientras que para la lógica difusa ,la regla de inferencia utilizado es el Modus Ponens Generalizado que dice[9]:

Premisa 1: "x es A*"

Premisa 2: "Si x es A entonces y es B"

Consecuencia : " y es B*"

Aquí A, B, A*, B* son conjuntos difusos.

Para entender mejor esta regla de inferencia ,se citará un ejemplo:

Considérese la regla "Si un hombre es pequeño, Entonces el no llegara a ser un muy buen jugador de basketball". Aquí hay el conjunto difuso A "es hombre pequeño", y el conjunto difuso B "no un muy buen jugador de basketball". Ahora se dará la premisa 1, como "Este hombre esta bajo los 1.50 metros de altura. Aquí A* es el conjunto difuso "hombre bajo el 1,50 metros", claramente se puede ver que $A^* \neq A$, pero son similares. De aquí se llegará a la siguiente consecuencia: "El será un mal jugador profesional de basketball". Por lo tanto B* es el conjunto difuso mal jugador profesional de basketball, y $B \neq B^*$, aunque claro está que son similares.

Al escribir el Modus ponens Generalizado de una forma más compacta sería[9]:

$$(x \text{ es } A^*) \wedge (x \text{ es } A) \Rightarrow (y \text{ es } B). \quad \text{expresión [1.30]}$$

Al tratar la expresión [1.30] de generalizar hacia la teoría de conjuntos, sería normal reemplazar las relaciones en vez de proposiciones ;de este razonamiento llegamos a la conclusión , el modus ponens generalizado resulta ser una operación de composición[9].

Como se trata de conjuntos difusos, la composición a utilizarse es la supremo asterisco, por lo tanto de la expresión [1.23] se llega a:

$$\mu_{B^*}(y) = \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \Rightarrow B}(x,y)] \quad \text{expresión [1.31]}$$

Con el propósito de entender mejor esta expresión[1.31], asumiremos que $\mu_{A^*}(x) = 1$ para $x=x'$ y $\mu_{A^*}(x) = 0$ para cualquier otro x que pertenezca a U , a esto se le denomina fusificador singleton[9]. En consecuencia:

$$\begin{aligned}
 \mu_{B^*}(y) &= \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \Rightarrow B}(x,y)] \\
 &= \mu_{A^*}(x') * \mu_{A \Rightarrow B}(x',y) \\
 &= 1 * \mu_{A \Rightarrow B}(x',y) = \text{mínimo} [1, \mu_{A \Rightarrow B}(x',y)] \\
 &= \mu_{A \Rightarrow B}(x',y) = 1 - \text{mínimo}[\mu_{A}(x'), 1 - \mu_{B}(y)] \text{ esta}
 \end{aligned}$$

será la expresión [1.32].

Este mismo resultado se hubiese obtenido si se utilizaba el producto como la norma t para la operación supremo asterisco, siempre y cuando se utilice un fusificador singleton

Una interpretación gráfica de este resultado es mostrado en la figura 1.8, como se puede apreciar esto viola el sentido común para una aplicación de ingeniería.

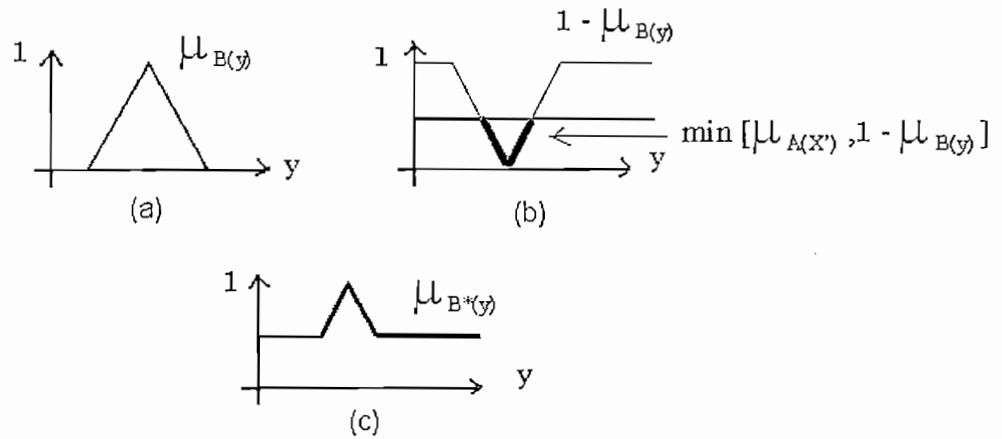


Figura 1.8. Construcción de $\mu_{B*(y)}$

Fue Mandani el que reconoció este problema ,debido a esto el dedujo su implicación, conocida con el nombre de implicación de Mandani [9] .

Implicación de Mandani:

$$\mu_{A \Rightarrow B}(x, y) = \text{mínimo}[\mu_{A(x)}, \mu_{B(x)}] \text{ expresión [1.33]}$$

Sus razones para elegir esta definición no ha sido basada en la causa y efecto, sino más bien en simplicidad de cálculo[9].

Después también fue Larsen quién dio otra definición de implicación[9]:

Implicación de Larsen:

$$\mu_{A \Rightarrow B}(x, y) = \mu_{A(x)} \cdot \mu_{B(x)} \text{ expresión [1.34]}$$

De esta manera con la implicación de Mandani (mínimo) y con la de Larsen (producto) no se llega a violar el sentido común de ingeniería.

Para demostrar lo antes dicho se parte de lo siguiente:

$$\begin{aligned}\mu_{B^*(y)} &= \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \Rightarrow B}(x, y)] \\ &= \mu_{A^*}(x') * \mu_{A \Rightarrow B}(x', y) \\ &= 1 * \mu_{A \Rightarrow B}(x', y) = \mu_{A \Rightarrow B}(x', y) \text{ expresión [1.35]}\end{aligned}$$

Primero consideremos la implicación de Mandani aplicando a la expresión [1.35] llegamos a:

$$\mu_{B^*(y)} = \text{mínimo}[\mu_{A(x')}, \mu_{B(y)}] \text{ expresión [1.36]}$$

Una representación gráfica de este resultado esta dado en la figura 1.9

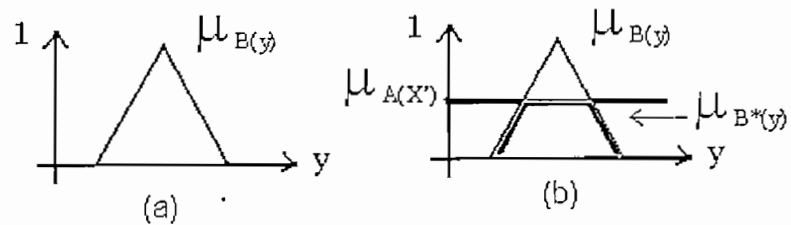


Figura 1.9. Implicación de Mandani.

Seguidamente consideremos la implicación de Larsen, aplicando a la expresión [1.35] nos conduce a:

$$\mu_{B^*(y)} = \mu_{A(x')} \cdot \mu_{B(y)} \text{ expresión [1.37]}$$

La interpretación gráfica de este resultado es mostrado en la figura (1.10)

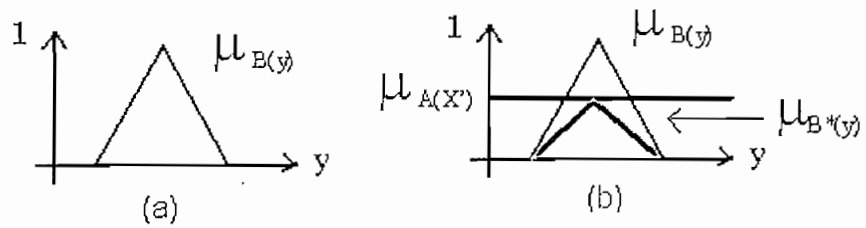


Figura 1.10. Implicación de Larsen.

CAPITULO 2

CONTROL DIFUSO

2.1 OBTENCION DE LA BASE DE CONOCIMIENTO

El diseño de controladores difusos requiere más pasos de diseño que lo usual, como por ejemplo de una base de reglas, mecanismo de inferencia, pre- y post procesamiento de datos.

Así como la lógica difusa puede ser descrita simplemente como “el cálculo con palabras que con números” [4], el control difuso puede ser descrito simplemente como “el control con oraciones que con ecuaciones”[4].

Un controlador difuso puede incluir reglas empíricas, como son utilizadas por el operador o el experto.

Un típico controlador difuso funciona empleando reglas lógicas con el formato .Si-Entonces, y formalmente el lado-Si es llamado la *condición* y el lado-Entonces es llamado la conclusión. Ejemplo:

- 1.Si error es Negativo y cambio de error es Negativo entonces salida Negativa Pequeña.
2. Si error es Negativo y cambio de error es Cero entonces salida Negativa Mediana.

La colección de estas reglas es llamada una base de reglas o base de conocimiento. La estrategia de control está basado en almacenar el lenguaje natural.

Existen algunas configuraciones para controladores difusos .Uno de los más utilizados es el control directo (figura 2.1), donde el controlador está en serie con la planta(o proceso), y realimentado con un lazo de control. Aquí la salida del proceso es comparada con una referencia(set point), y si existe una desviación, el controlador entra en acción acorde a la estrategia de control, este tipo de control puede remplazar al control convencional PID[4].

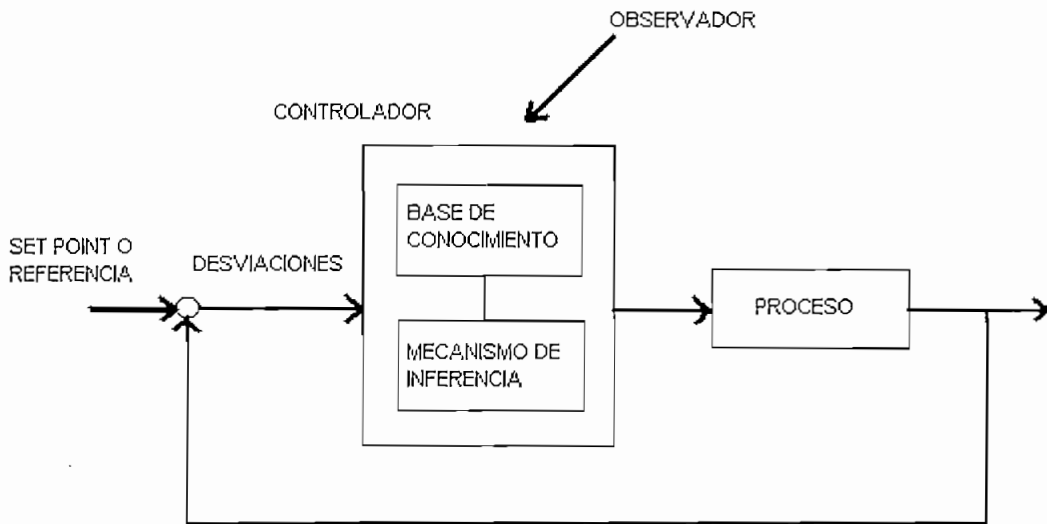


Figura.2.1.Control Directo.

Existen al menos cuatro fuentes principales para encontrar reglas de control (Takagi & Sugeno in Lee, 1990).

Experiencia del experto y conocimiento de ingeniería de control

Un ejemplo clásico es el manual del operador para una planta de cemento (Holmblad & Ostergaard, 1982).Consiste en una colección de reglas, que preguntan al experto o al operador usando un cuestionario cuidadosamente organizado.

Basada en las acciones de control del operador

Reglas difusas Si-Entonces pueden ser deducidas de las observaciones de las acciones de control del operador. Estas reglas expresan una relación

entre las entradas y las salidas.

Basados en un modelo difuso del proceso.

Una base de reglas lingüísticas puede ser visto como un modelo inverso del proceso controlado. Este método está restringido únicamente para sistemas de bajo orden .

Basados en el aprendizaje.

Este es un ejemplo de controladores que encuentran sus propias reglas. Redes Neuronales es otra posibilidad.

2.2 ESTRUCTURA DE UN CONTROLADOR DIFUSO

En el numeral anterior se indicó una configuración típica de un controlador difuso, pero existen componentes específicos que caracterizan a un controlador difuso para realizar el diseño. En el diagrama de bloques de la figura 2.2, el controlador se halla entre el bloque de pre-procesamiento y un bloque de post-procesamiento. Los componentes se explican con más detalles a continuación:

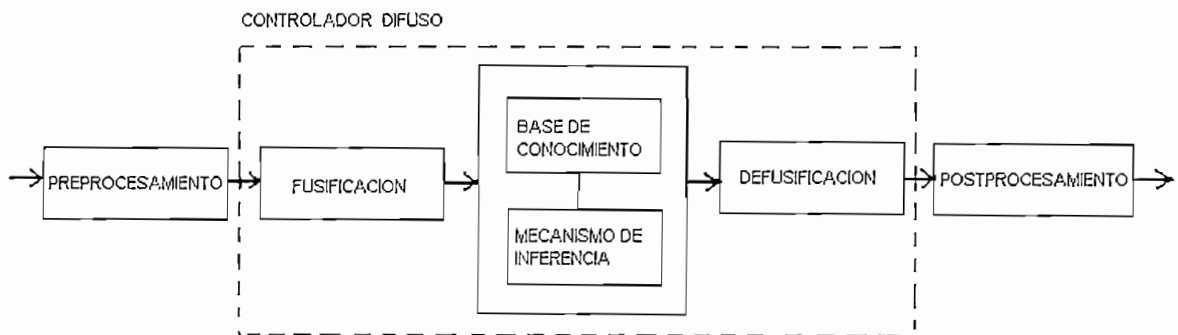


Figura2.2. Bloques de un controlador difuso.

2.2.1 PREPROCESAMIENTO

Un pre-procesador, el primer bloque de la figura 2.2, acondiciona de la mejor manera la señal, antes de que ella entre al controlador. Ejemplos donde se necesiten de pre-procesamiento son:

- Cuantización con la muestra o redondeo a enteros;

Normalización o escalamiento para llegar a un rango estándar; que puede ser uniforme, no uniforme, o las dos cosas. Se muestra un ejemplo general en la figura 2.3 donde el universo del discurso, $[-6.0, +4.5]$, es transformado dentro de un intervalo normalizado $[-100, +100]$.

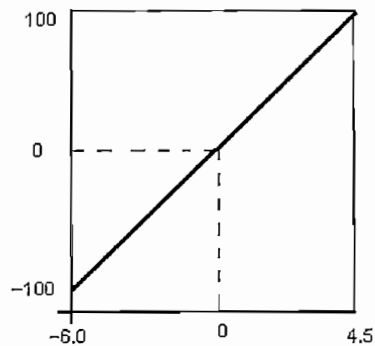


Figura.2.3.Ejemplo de un escalamiento no lineal de una medida de datos.

- Filtrado de la señal de manera de eliminar el ruido, pues las entradas son a menudo distorsionadas por el ruido.

2.2.2 FUSIFICACION

Luego del bloque de pre-procesamiento, el primer bloque dentro del controlador es la que realiza la fusificación, la cual convierte cada dato de la señal de entrada a grados de pertenencia, por medio de una o algunas funciones de pertenencia a las cuales se las etiqueta.

2.2.3 BASE DE CONOCIMIENTO

Un sistema difuso está caracterizado por un conjunto de términos lingüísticos basados en el conocimiento del experto. Estas reglas además pueden ser aplicadas tanto a un problema MIMO (múltiple entradas y múltiples salidas) como a problemas SISO (una entrada y una salida). El típico problema SISO consiste en regular una señal de control basada en el error y el cambio de error de la

señal, aunque en otros casos también necesite de la acumulación de error como entrada.

También existe otro formato de agrupar todas estas reglas, que consiste en introducirlas en una matriz, a este arreglo se le denomina matriz FAM (Memoria Asociativa Difusa) que se muestra en al figura 2.4. A continuación se muestra esta otra forma de representar a este grupo de reglas.

		CAMBIO DEL ERROR		
		Neg	Cero	Pos
ERROR	Neg	NB	NM	CERO
	Cero	NM	CERO	PM
	Pos	CERO	PM	PB

Figura 2.4. Matriz de memoria asociativa difusa.

Ahora véase , la figura 2.5 muestra la respuesta del sistema de un proceso a ser controlado, donde las variables de entradas a un controlador difuso son el error y el cambio de error y la salida es CI. Se asume que el número de funciones son en igual número para las entradas y las salidas, con comunes etiquetas (negativo, cero, positivo). La justificación de las reglas del control difuso se pueden explicar en base a las regiones , así con regla correspondiente de la región i puede ser formulada como una regla R_i y tiene el efecto de disminuir el tiempo de

risado. Mientras que la regla Rii para la región ii decrementa el sobre-impulso de la respuesta del sistema [2]. Mas específicamente:

Ri : Si (E es positivo y CE es negativo) entonces CI es positivo.

Rii : Si (E es negativo y CE es negativo) entonces CI es negativo.

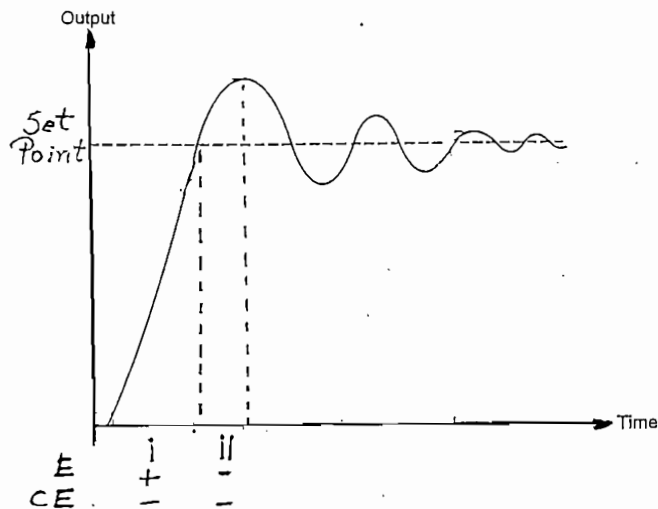


Figura.2.5.Respuesta de un sistema a un escalón.

Mejores características de control puede ser obtenido usando finas particiones en el universo del discurso, esto quiere decir al aumentar el numero de funciones de pertenencia, por ejemplo con el conjunto de funciones { NB, NM, NS, ZE , PS, PM, PB}.

Al realizar el diseño cabe mencionar con respecto a las funciones de pertenencia, que existen algunos criterios para seleccionar que tipo de función de pertenencia se ha de utilizar, estos criterios son:

- La función de pertenencia debería ser lo suficientemente ancha para tener en cuenta el ruido.

- Se recomienda una cierta cantidad de solapamiento; de otra manera se desempeñará muy pobremente el controlador.
- El ancho de las funciones de pertenencia depende del número de funciones y viceversa.

Otros criterios válidos son también :

- *Se debe empezar con funciones de pertenencia triangulares.* Todas las funciones para una particular entrada o salida , debería ser triángulos simétricos del mismo ancho.
- *El solapamiento debería ser del 50%.* Los anchos deberían ser inicialmente elegidos de manera que cada valor del universo es socio o parte de al menos dos funciones de pertenencia ,excluyendo de esta posibilidad para los elementos en los extremos finales.

2.2.4 MECANISMO DE INFERENCIA

El mecanismo de inferencia empleado en un controlador difuso es generalmente mucho más simple que aquellos usados en un sistema típico experto, pues en un controlador difuso el consecuente de una regla no es aplicado al antecedente de otra.

En las figuras 2.6 (a) y 2.6 (b) se describe de una manera más detallada el mecanismo de inferencia; es decir como se aplica el razonamiento para decidir en base a reglas a fin de llegar a una conclusión lógica. Así la figura 2.6 (a) y 2.6 (b) en la que se tiene dos entradas y una salida u , v , w respectivamente.

Ahora bien, al entrar en un instante dos datos, u_0 para u , y v_0 para v , estos valores pasaran a tener su grado de pertenencia dentro de las funciones de

pertenencia A_1, A_2 (para u_0) y B_1, B_2 (para v_0), por lo anterior se observará que se activan cuatro reglas:

1. Si u_0 es A_1 y v_0 es B_1 entonces la salida es C_1 ;
2. Si u_0 es A_1 y v_0 es B_2 entonces la salida es C_2 ;
3. Si u_0 es A_2 y v_0 es B_1 entonces la salida es C_1 ;
4. Si u_0 es A_2 y v_0 es B_2 entonces la salida es C_2 ;

En los gráficos 2.6 (a) y 2.6 (b) se tomarán en cuenta solamente para motivos de ilustración las reglas 1 y 4.

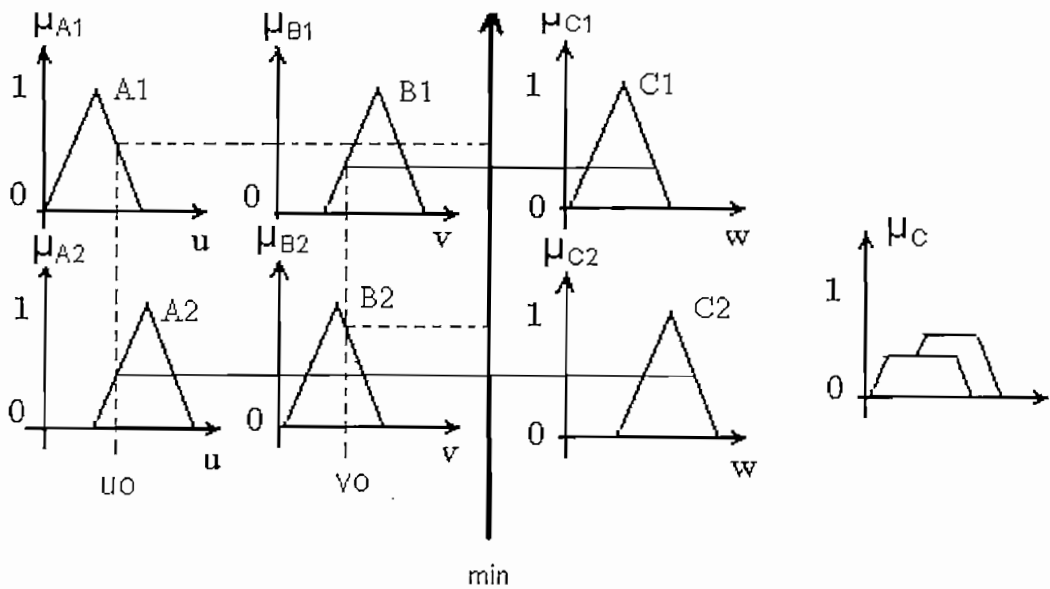


Figura 2.6 (a) se usa el mínimo para la intersección y para la implicación el mínimo de Mandani.

Hay que notar del gráfico anterior que las funciones de pertenencia A_1 y A_2 están en un mismo universo de discurso al igual que las funciones B_1, B_2 , pero por motivos de explicación se encuentran separadas.

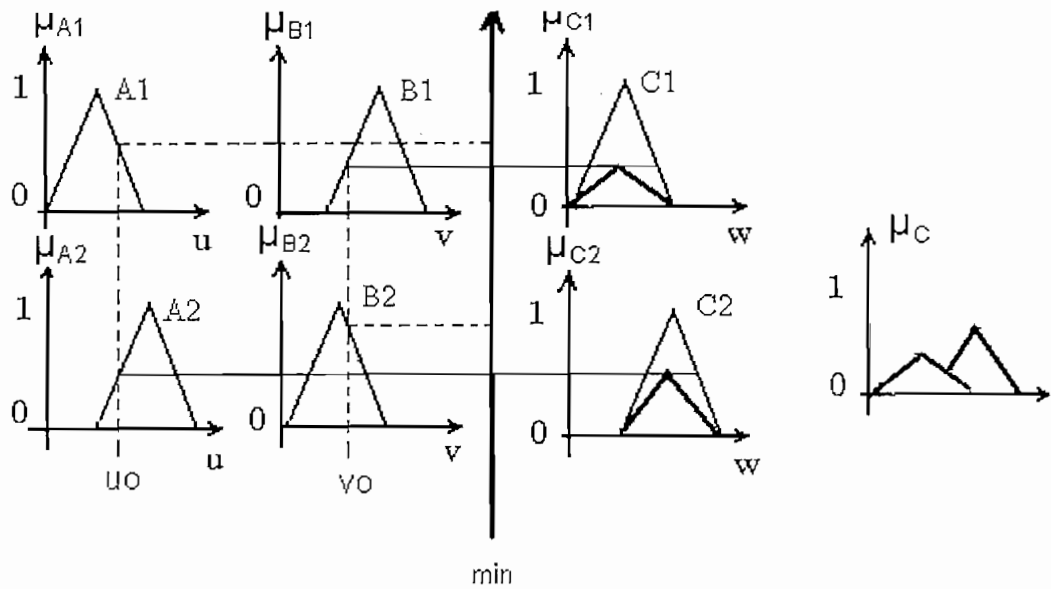


Figura 2.6(b) se usa el mínimo para la intersección y para la implicación el producto de Larsen.

Se notará que en ambos casos las áreas se suman, por consiguiente cuando se activan cuatro reglas se deberán sumar las cuatro áreas ,una por cada regla.

2.2.5 DEFUSIFICACION

Básicamente la defusificación consiste en un mapeo de las acciones de control difuso definidas sobre un universo del discurso dentro de acciones de control en un espacio no difuso [2].

De las áreas que se obtuvieron de las figuras 2.6 (a) y 2.6 (b) deberán ser convertidas a números, para que puedan ser enviadas al proceso como una señal de control. Para este propósito, existen muchos métodos de defusificación entre los más importante tenemos:

Centro de Gravedad (COG) .

Este método calcula el centro de gravedad del área de las funciones de membresía inferidas. La ecuación general del COG es la siguiente:

$$Y = \frac{\sum c_i \int \mu_{Bi}(y) \cdot dy}{\sum \int \mu_{Bi}(y) \cdot dy} \quad \text{expresión [2.1]}$$

Donde:

c_i es el valor para la cual la función de pertenencia alcanza su valor máximo.

$\int \mu_{Bi}(y) \cdot dy$ es el área debajo de la función $\mu_{Bi}(y)$.

Como condición necesaria para aplicar este método, se debe cumplir que el denominador de la expresión [2.1] debe ser diferente de cero.

Centro Promedio (COA)

La salida de control es calculada utilizando el centro de cada una de las funciones de pertenencia.

$$Y = \frac{\sum c_i \cdot \mu_{Bi}(y)}{\sum \mu_{Bi}(y)} \quad \text{expresión [2.2]}$$

Donde :

c_i es el centro de las funciones de pertenencia.

$\mu_{Bi}(y)$ es el valor del grado de pertenencia que se activa.

Además de estos métodos existen otros que se pueden ver en [4].

2.2.6 POSTPROCESAMIENTO

El escalamiento a la salida también es relevante. En el caso que la salida es definida en un universo estándar, éste debería luego ser escalado a unidades de

ingeniería, como a voltios, metros, etc. Un ejemplo es el escalamiento de un universo estándar $[-100, 100]$ a unidades físicas $[-10, 10]$ voltios.

CAPITULO 3

DISEÑO DEL CONTROLADOR

3.1 DISEÑO DEL HARDWARE

3.1.1 SENSORES DE NIVEL DE LIQUIDO

Existe una variedad de sensores de nivel de líquidos que aprovechan las características eléctricas y físicas en sus principios de transducción.

Entre los sensores de nivel de líquido más utilizados tenemos: capacitivos, resistivos, inductivos, de presión diferencial.

Idealmente el sensor de nivel de líquido que se debe utilizar para este tipo de control deberá tener una respuesta muy buena (lineal), y una buena característica de inmunidad al ruido, no obstante, se pueden superar las deficiencias en este sentido.

El sensor que se ha utilizado es el proporcionado por el laboratorio de Control Automático, el sensor de presión diferencial de estado sólido (SCX01DNC de la fábrica SENSYM) de medición indirectamente .

Lo que mide este tipo de sensor es la presión hidrostática en el fondo del tanque, ya que esta variable es directamente proporcional a la altura del líquido que la produce.

Se ha utilizado la configuración diferencial, ya que en el fondo del tanque lo que se tiene es en definitiva la suma de la presión hidrostática producida por la columna de agua, más la presión atmosférica, por lo tanto es necesario restar la atmosférica. En la figura 3.1 se ilustra lo antes dicho.

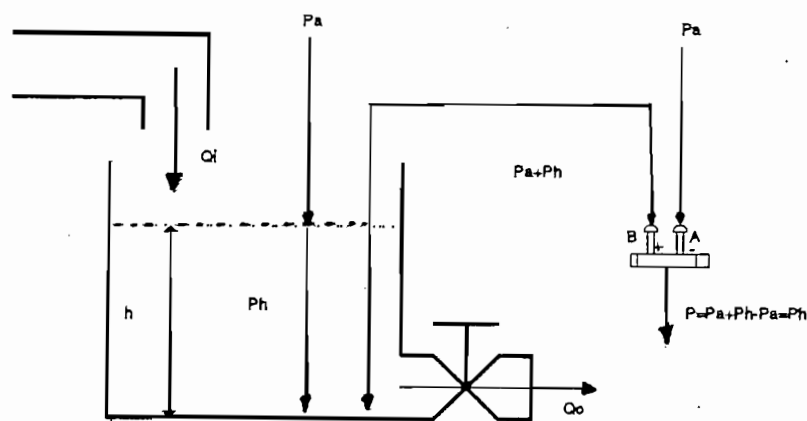


Figura 3.1. Configuración en modo diferencial para sensor el nivel de líquido.

En esta figura:

P_a = presión atmosférica

P_h = presión hidrostática = $\delta \cdot g \cdot h$

δ = densidad del agua [1 Kg/cm³]

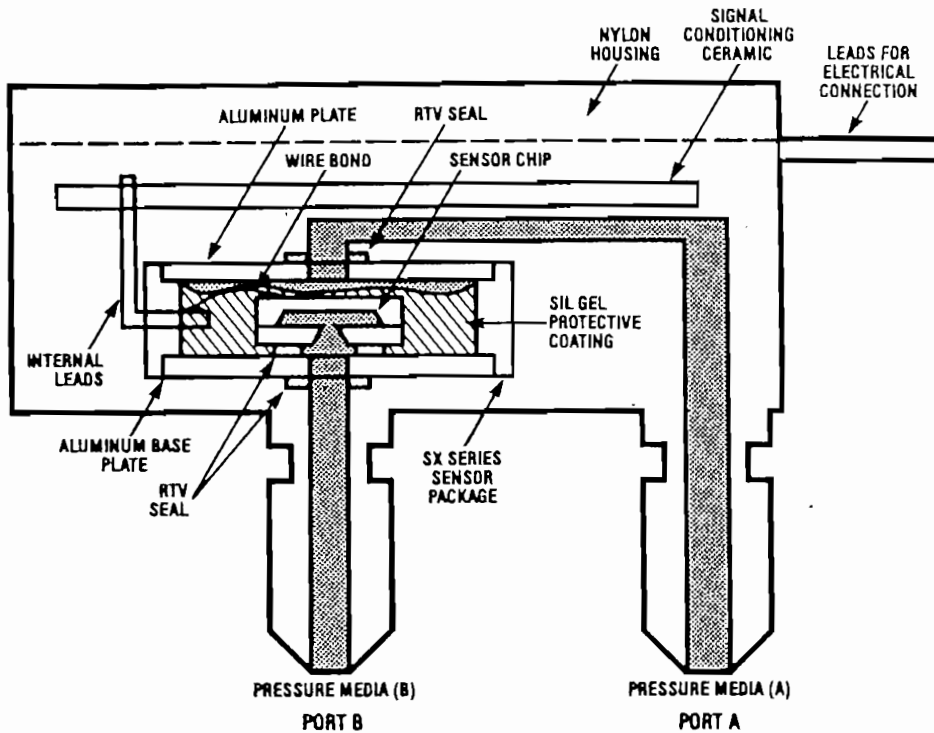
h = altura del agua dentro del tanque.

El rango de temperatura de este sensor es de 0°C a 70°C, y además trabaja con fluidos gaseosos no corrosivos (aire y gases secos, agua).

Además el sensor está conformado por un circuito integrado y una delgada película cerámica cortada a láser, empacados dentro de un encapsulado de nylon para evitar daños debido a fuerzas externas .

El encapsulado tiene dos pórticos o tubos de presión montados ,compatibles con tubos de plásticos estándares para las conexiones de presión. Para esta

aplicación, el p $\acute{o$ rtico A debe estar siempre abierto a sensor la atm $\acute{o$ sfera, mientras que el p $\acute{o$ rtico o tubo B se debe conectar la manguerita de pl \acute{a} stico que est \acute{a} acoplada a la sonda de vidrio que est \acute{a} introducida en el fondo del tanque, ver figura 3.2



Physical Construction (Cutaway Diagram)

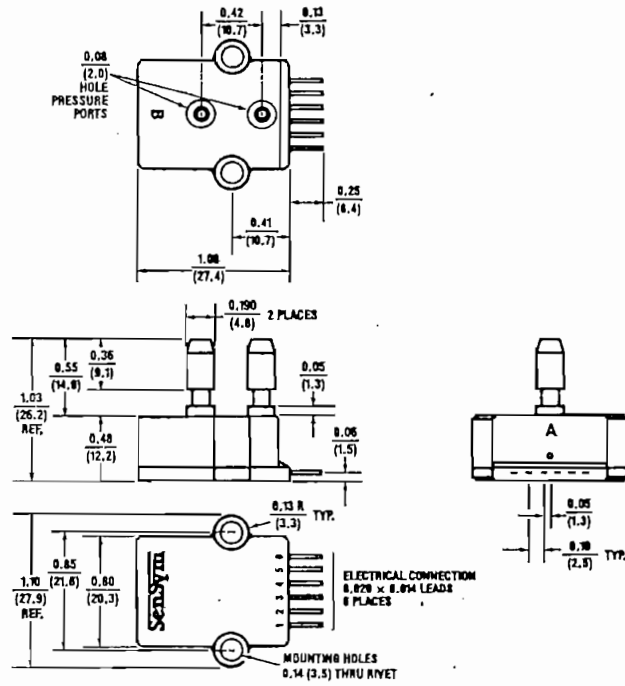
(Not Drawn to Scale)

Figura 3.2. Construcci $\acute{o$ n f $\acute{i$ sica del sensor.

La figura 3.3 muestra las dimensiones f $\acute{i$ sicas del sensor, y la figura 3.4 la tarjeta de acondicionamiento, sobre la que va montado el elemento.

PHYSICAL DIMENSIONS

SCX C Series



DIMENSIONS IN INCHES (MILLIMETERS)
WEIGHT: 3 GRAMS
CASE MATERIAL: GLASS FILLED NYLON

WETTED MATERIAL
PORT A: GLASS FILLED NYLON, RTV, SILICEL
PORT B: GLASS FILLED NYLON, SILICON, RTV
SEE PHYSICAL CONSTRUCTION DRAWINGS

Figura 3.3. Dimensiones físicas del sensor.

PHYSICAL DIMENSIONS

SCX Evaluation Board

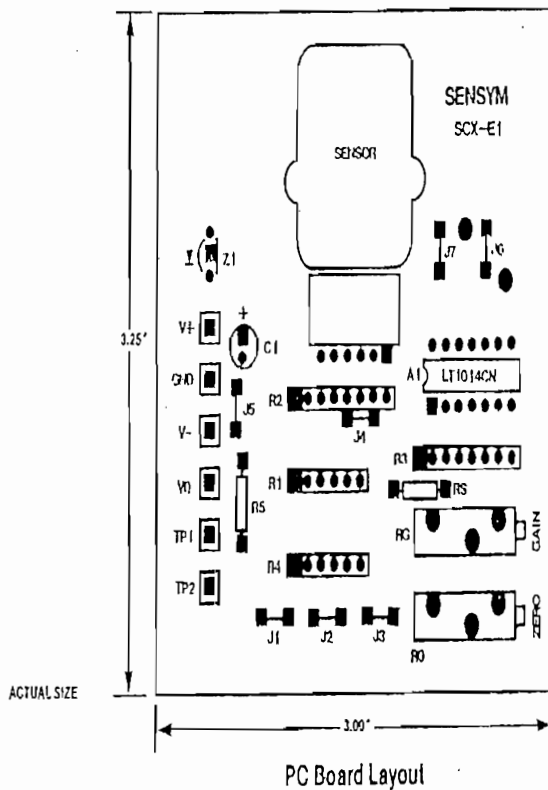


Figura 3.4. Construcción física del sensor.

3.1.2 CARACTERÍSTICAS DEL SENSOR Y ACONDICIONAMIENTO

A continuación se presenta las características del sensor:

TABLA 4.1 CARACTERÍSTICAS DE OPERACIÓN DEL SENSOR SCX01DNC

CARACTERÍSTICAS	MINIMO	TIPICO	MAXIMO	UNIDAD
Rango de presión Operación		-	1	Psi
Sensibilidad	-	18	-	Mv/psi
Rango a plena escala	17.00	18.00	19.00	Mv
Offset de presión Cero	-1.0	0	+1.0	Mv
Histéresis y linealidad combinada	-	+/-0.2	+/- 1.0	%FSO
Efecto de la temperatura en el rango de (0-70°C)	-	+/-0.4	+/-0.2	%FSO
Efecto de la temperatura en el Offset (0-70°C)	-	+/-0.2	+/-0.1	mv
Repetibilidad	-	+/-0.2	+/-0.50	%FSO
Impedancia de Entrada	-	4.0	-	K Ω
Impedancia de salida	-	4.0	-	K Ω
Voltaje en modo común	5.7	6.0	6.3	Vdc
Estabilidad de Offset y rango	-	+/-0.1	-	%FSO

Se escogió el rango de trabajo del sensor de 0 a 1 psi es la presión hidrostática ejercida por una columna de 70.308 cm de agua, como en este caso el prototipo tiene una altura máxima de 40 cm, este rango de trabajo estandarizado es el mas adecuado. La respuesta de voltaje de acondicionamiento del sensor es de 0.25 Voltios/cm.

3.1.3 TARJETA DE ADQUISICIÓN DE DATOS

La tarjeta de adquisición de datos es la Lab-PC-1200 , tiene las siguientes características:

- Posee ocho canales analógicos de entrada que pueden configurarse como ocho entradas simples o como cuatro entradas diferenciales. con los siguientes rangos de voltaje:
Rango bipolar : ± 5 V.
Rango unipolar : 0 a 10 V.
- Dos canales analógicos de salida que pueden configurarse para trabajar con los siguientes modos de voltaje y corriente:
Rango bipolar : ± 5 V
Rango unipolar : 0 a 10 V.
Máxima corriente de carga : ± 2 mA
- Veinticuatro líneas de entradas/salidas digitales compatibles con señales de TTL y distribuidos en tres canales digitales: PA, PB, PC .
- Tres contadores/temporizadores de 16 bits.
- Un conector de entradas/salidas de 50 pines, cuya distribución se muestra en la figura 3.5.

ACH0	1	2	ACH1
ACH2	3	4	ACH3
ACH4	5	6	ACH5
ACH6	7	8	ACH7
AISENSE/AGND	9	10	DAC0OUT
AGND	11	12	DAC1OUT
DGND	13	14	PA0
PA1	15	16	PA2
PA3	17	18	PA4
PA5	19	20	PA6
PA7	21	22	PB0
PB1	23	24	PB2
PB3	25	26	PB4
PB5	27	28	PB6
PB7	29	30	PC0
PC1	31	32	PC2
PC3	33	34	PC4
PC5	35	36	PC6
PC7	37	38	EXTTRIG
EXTUPDAT*	39	40	EXTCONV*
OUTB0	41	42	GATB0
OUTB1	43	44	GATB1
CLKB1	45	46	OUTB2
GATB2	47	48	CLKB2
5V	49	50	DGND

Figura 3.5. Conector de la tarjeta Lab-PC _1200

En esta figura los pines son los siguientes:

ACH<0...7> = canales analógicos de entrada.

DAC0OUT = canal de salida analógica 0.

DAC1OUT = canal de salida analógica 1.

AGND = tierra analógica

DGND = tierra digital.

PA<0...7> = puerto digital A (bidireccional).

PB<0...7> = puerto digital B (bidireccional).

PC<0...7> = puerto digital C (bidireccional).

XXXB0 = señales de control y salida del contador B0.

XXXB1 = señales de control y salida del contador B1.
 5V = +5 Voltios

Como se puede observar, los terminales (pines) del conector se agrupan de la siguiente manera : terminales de entrada analógica, terminales de salida analógica, terminales de I/O digital, terminales para las señales de los temporizadores y pines de polarización.

A continuación se presentan características más importantes de las entradas y salidas analógicas ,pues fueron ellas las utilizadas.

Entrada Analógica:

Resolución	12 bits
Máxima velocidad de muestreo	100000 muestras/s
Protección de sobrevoltaje	+/- 35 V encendida +/- 25 V apagada
Precisión Relativa	Típico +/- 0.5 LSB Máximo +/- 1.5LSB
Impedancia de entrada	Encendido :100GΩ Apagada :4.7 KΩ Sobrecarga: 4.7KΩ

Tabla 3.1 Características de las entradas analógicas de la tarjeta Lab-PC-1200.

Salida Analógica:

Resolución	12 bits
Máxima velocidad de muestreo	1000 muestras /s,
Protección	Corto circuito a tierra
Precisión Relativa	Típico +/-0.25 LSB Máximo +/-0.5 LSB

Tabla 3.2. Características de las salidas analógicas de la tarjeta Lab-PC-1200.

Para este proyecto se ha utilizado una entrada y una salida analógica ,las dos están configuradas para 0-10V.

3.1.4 ACONDICIONAMIENTO DE LA SEÑAL

En este sistema , el agua es llevada desde el reservorio hacia el tanque ,esta acción se lleva a cabo por medio de una bomba centrífuga sumergible accionada por un motor de DC.

Las especificaciones más relevantes de esta bomba son:

Marca : WHIRLWIND

Modelo: 3BPLA 12 Vdc.

$V_{nom} = 12$ (Vdc)

$I_{nom} = 1.8$ (A)

Descarga nominal : 6 gpm.

Para la implementación de este tipo de control, se tuvieron que diseñar dos circuitos: uno de control y otro de potencia; ambos circuitos en conjunto, lo que

hacen es controlar la velocidad del motor de la bomba en proporción directa al voltaje de entrada proveniente de la tarjeta de adquisición de datos.

Para este propósito se utilizó un TROCEADOR DE DC, para la señal continua de potencia, y enviar solamente una fracción de la misma ,con lo que se consigue variar el voltaje de alimentación a la bomba en forma proporcional al voltaje de entrada al circuito de control.

El circuito de potencia diseñado , se ilustra en la figura 3.6, el mismo que está conformado por un transistor de potencia (interruptor estático), cuya función es la de conectar y desconectar alternativamente la fuente de alimentación de corriente continua, produciendo sobre la carga un voltaje troceado. En serie con este transistor de potencia , se encuentra a un fusible de protección contra sobrecarga de corriente de 2 Amperios . Mientras que el diodo D en paralelo con la bomba, es un diodo de conmutación y sirve como camino alternativo para la corriente almacenada por la carga inductiva del motor de la bomba, cuando el transistor se abre.

Nótese que las referencias del circuito de control (generador de la señal PWM) , y el circuito de potencia no son las mismas, pues la señal proveniente del circuito de control entra a la base del transistor esta referida al cátodo del diodo, y no a la tierra del circuito de potencia.

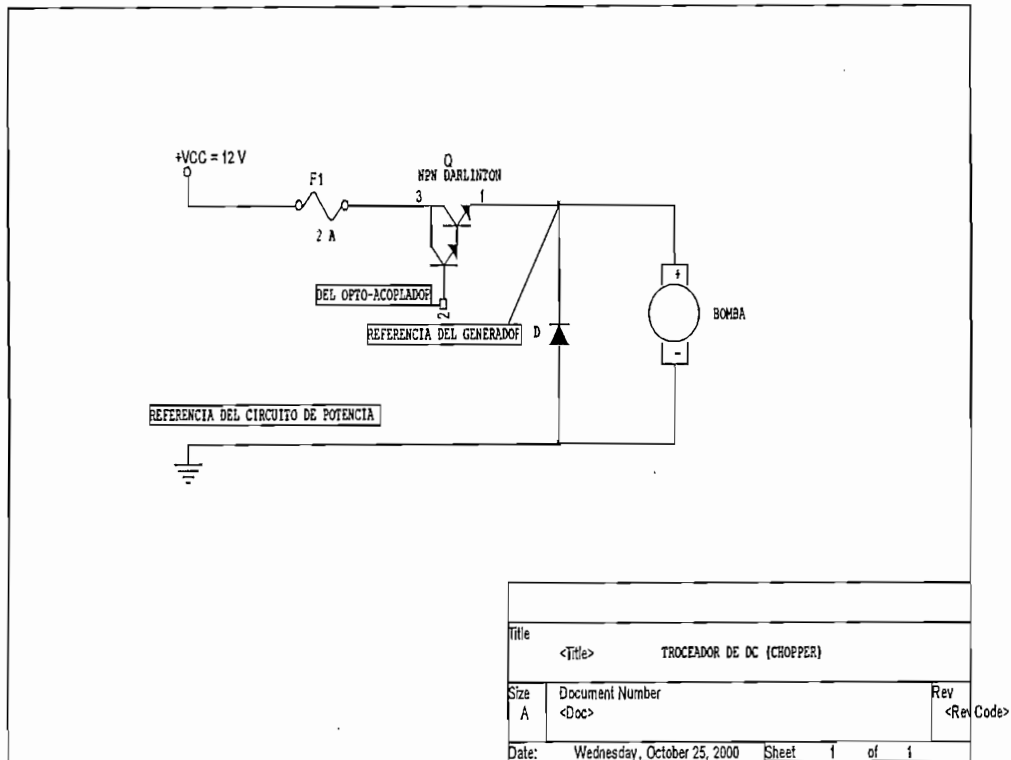


Figura.3.6.Circuito de potencia utilizado.

Para el diseño de circuito de potencia se toma en cuenta la corriente nominal de la carga(bomba de voltaje continuo tipo sumergible), con lo que se deduce que :

I diodo > 2 Amperios

I transistor > 2 Amperios

La técnica de modulación por ancho de pulso (PWM), se utiliza para controlar la apertura y cierre del transistor de potencia , esta señal PWM es generada por el circuito integrado LM3524 con su respectiva circuitería adicional como se muestra en la figura 3.7. El LM3524 genera una señal modulada por ancho de pulso cuya frecuencia de trabajo está determinada por el capacitor C2 y el potenciómetro P, los valores de estos elementos se determinaron experimentalmente basadas en las curvas dadas por el fabricante .Las resistencias R1,R2,R3 y R4 en paralelo y las resistencias R5 se usan para limitar la corriente en la base del transistor de

potencia que controla a la bomba. Mientras que el capacitor C1 sirve para filtrar cualquier señal de ruido, estos valores se los encuentra en los manuales del fabricante.

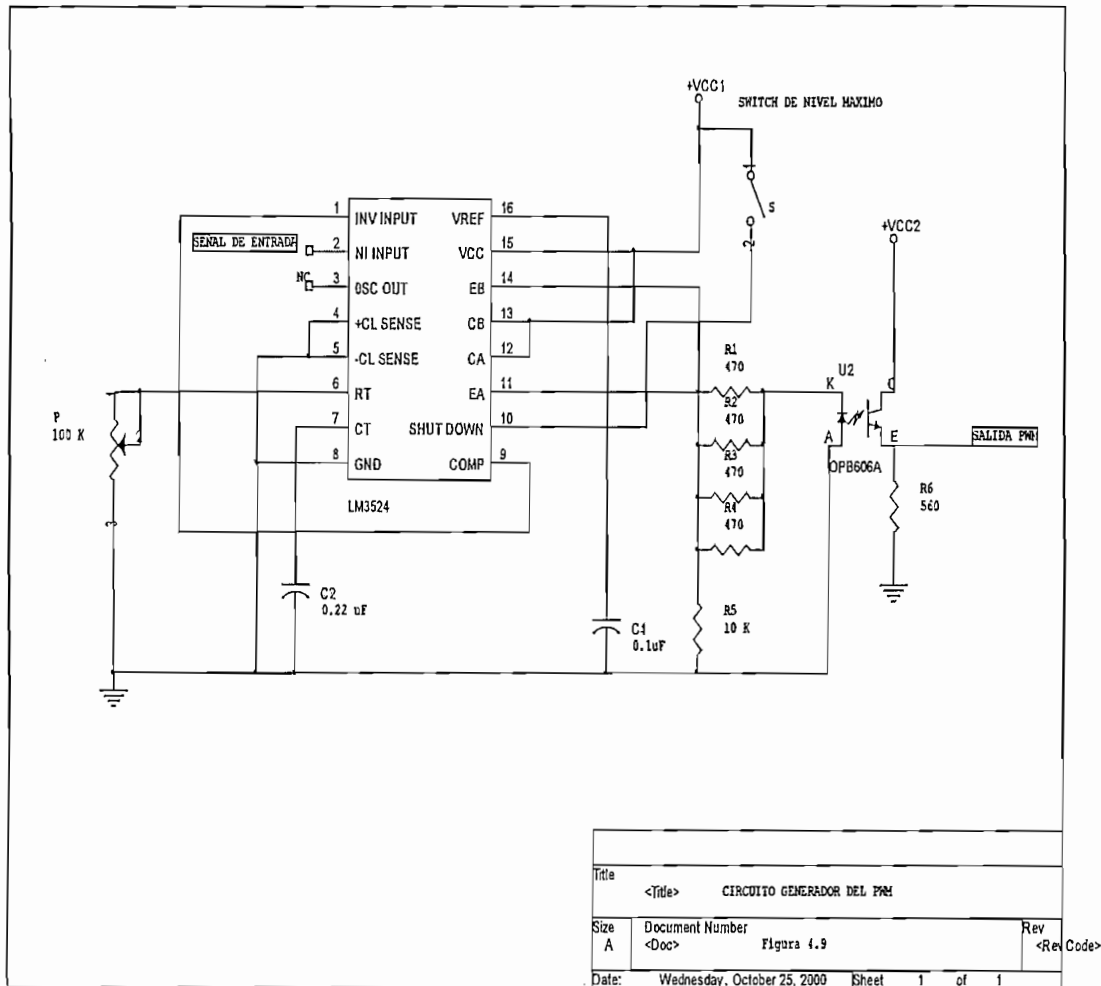


Figura 3.7.Circuito de control

3.1.5 FUENTES DE ENERGIA UTILIZADAS

Para el funcionamiento del sistema se utilizan tres fuentes de energía, dos fuentes para el circuito de control (se utilizan opto - acopladores para aislar las tierras , para protección de la tarjeta de Labview) y una fuente de poder para alimentar a la bomba tipo sumergible.

A continuación se describen las características de las fuentes utilizadas :

Fuente para el circuito de Control

1 fuentes de voltaje continuo regulables , proporcionados por el laboratorio de Control Automático, de la que se obtienen 2 valores de voltaje independientes.

Fuente para el circuito de Potencia

Potencia : 73 W

Voltaje de Entrada : 115/230 Vac 40/50 Hz.

Voltaje de Salida dc : +5V/5A +12 V/4A.

3.2 DISEÑO DEL SOFTWARE

3.2.1 ACERCA DE LOS PROGRAMAS UTILIZADOS

Para la realización del software requerido se utilizó el Turbo C++ y Labview.

Turbo C++ es utilizado para realizar el programa de control propiamente dicho, y es aquí donde se realiza el proceso de inferencia.

Labview es utilizado para la adquisición de datos, filtrado de la señal ,escalonamiento de la señal tanto a la entrada como a la salida ,y la salida de la señal a través de la tarjeta hacia el circuito de control.

3.2.2 CONSTRUCCION DE LIBRERIAS DINAMICAS

Para este efecto, Labview adquiere datos a través de la tarjeta respectiva, y con una de sus herramientas avanzadas :Call Library Function, se puede acceder a una DLL (Dinamic Link Library) programa desarrollado en C y compilada en Visual C++ para obtener la DLL, la cual tiene la propiedad de compartir con Labview los valores de entrada y salida de datos.

Para entender de mejor manera la construcción de estas DLLs ,se presenta el siguiente ejemplo:

Supóngase que se tiene un programa escrito en C++, para la suma de tres números:

```
#include<stdio.h>
float coef1,coef2,coef3;
float main(void)
{
return coef1+coef2+coef3;
```



```
}
```

Lo primero que se tiene que realizar es buscar la librería de Labview llamada Extcode.h, y copiarla a la librería BIN de Visual C++, para que exista la interacción de estos dos programas.

Al código anterior se deben añadir las siguientes instrucciones y librerías para poder ser importado y compilado en VISUAL C++, de esta manera quedaría de la siguiente forma:

```
#include<stdio.h>
#include<extcode.h>
#include<windows.h>
BOOL WINAPI DllMain( HANDLE hDll, DWORD dwReason, LPVOID
lpReserved)
{
Return TRUE;
}
__declspec(dllexport) float suma(float coef1, float coef2, float coef3)
{
return coef1+coef2+coef3;
}
```

El siguiente y último paso es importar esta DLL a Labview, mediante su herramienta CALL LIBRARY FUNCTION, esta herramienta propia de labview debe ser configurada de acuerdo al programa realizado en C++.

3.2.3 PASOS PARA LA IMPLEMENTACION

El primer paso antes de calcular el error y la tasa de cambio del error consiste, en mejorar la señal proveniente del sensor para esto, se ha utilizado un filtro digital pasabajo de Butterword de segundo orden, con frecuencia de muestreo $f_s = 2.035$

Hz, y con frecuencia de corte $f_c = 0.15$ Hz, la obtención de estos parámetros se lo realizó experimentalmente, teniendo siempre presente el teorema del muestro (criterio de Nyquist), que afirma:

$$0 < f_{\text{corte}} < 0.5 * f_{\text{muestreo}}. \quad \text{Expresión [3.1]}$$

Ahora bien, para realizar el controlador difuso es indispensable tener las señales del error y del cambio del error (tasa de cambio del error) como entradas al controlador y que se describen como:

$$E = r - y \quad \text{expresión [3.2]}$$

Donde:

E = error

r = Referencia o Set point.

y = medida de la variable a controlar.

$$\Delta E = E_n - E_{n-1} \quad \text{expresión [3.3]}$$

Desarrollando la expresión [3.3] llegamos a:

$$\Delta E = Y_{n-1} - Y_n \quad \text{expresión [3.4]}$$

De lo que se llega a deducir, se puede llegar a obtener errores y cambio de errores :positivos, negativos, y cero.

En las figuras 3.8,3.10,3.12 ,3.14 se puede observar la señal sin filtrar, mientras que en las figuras 3.9,3.11,3.13,3.15 las señales se han mejorado considerablemente por efecto del filtro.

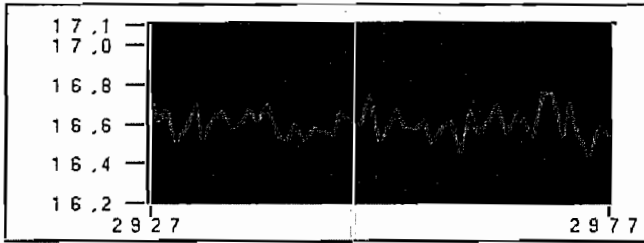


Figura 3.8. Señal del sensor sin filtrar.

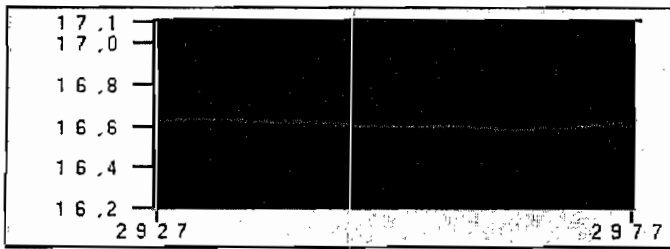


Figura 3.9 Señal filtrada de la figura 3.8 .

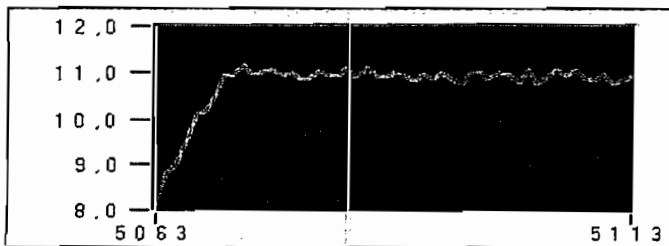


Figura.3.10. Señal del sensor sin filtrar.

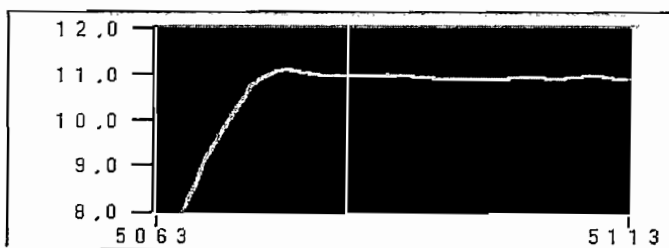


Figura 3.11. Señal filtrada de la figura 3.10.

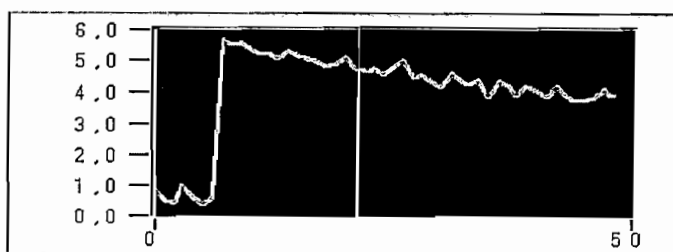


Figura 3.12. Señal proveniente del sensor sin filtrar.

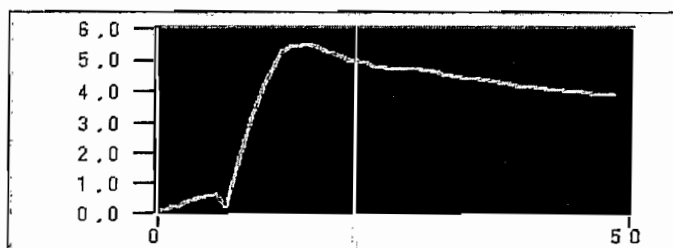


Figura 3.13. Señal filtrada de la figura 3.12.

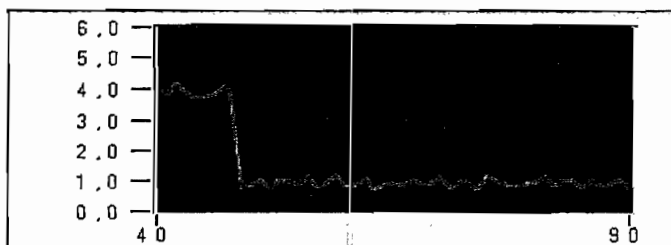


Figura 3.14. Señal proveniente del sensor sin filtrar.

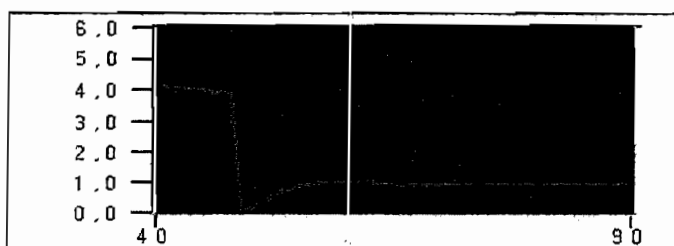


Figura 3.15. Señal filtrada de la figura 3.14

Las gráficas anteriores se ha observado, que los filtros eliminan considerablemente el ruido que contamina a la señal, y que ayuda a la determinación correcta de la tasa de cambio.

Con la misma señal filtrada, procedemos a determinar experimentalmente el universo del discurso del error, de su tasa de cambio, y de la salida , con el propósito de fusificar estas señales.

A continuación se dan los rangos del universo de discurso del error, cambio del error, según los valores medidos resultan:

Error	[-38	38]
Δ Error	[-0.6	0.2]

En cambio, el rango del universo del discurso de la salida, es proporcionado por la salida del circuito de control :

Salida	[1.6	3.4]
--------	------	------

Si se particiona con 7 funciones de pertenencia triangulares, dentro de estos rangos de los universos del discurso, de cada variable ,con las siguientes etiquetas :

NG = Negativo Grande.

NM =Negativo Mediano.

NP = Negativo Pequeño.

ZR = Cero.

PP = Positivo Pequeño.

PM = Positivo Mediano.

PG = Positivo Grande

Se llegará a obtener lo siguiente:

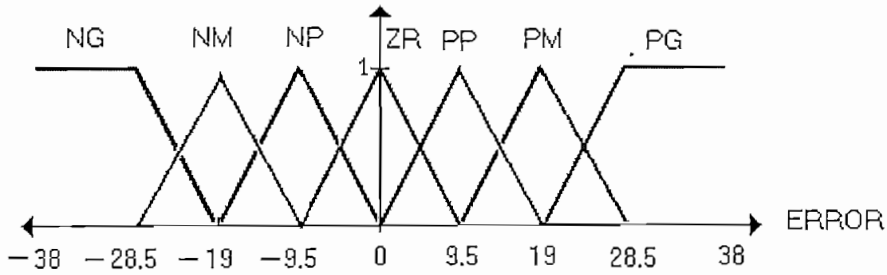


Figura 3.16. Fusificación del Error.

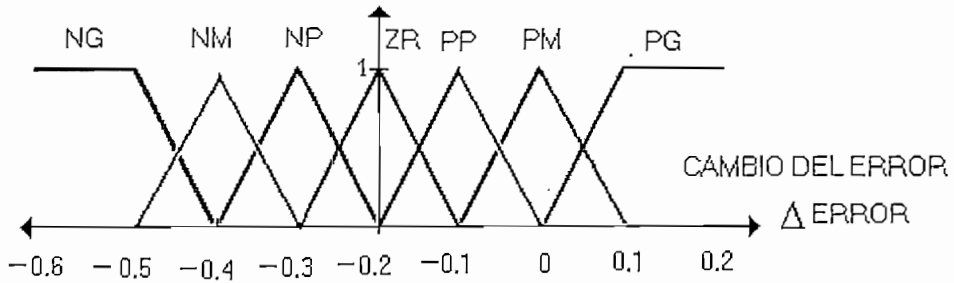


Figura 3.17. Fusificación de la tasa de cambio del error.

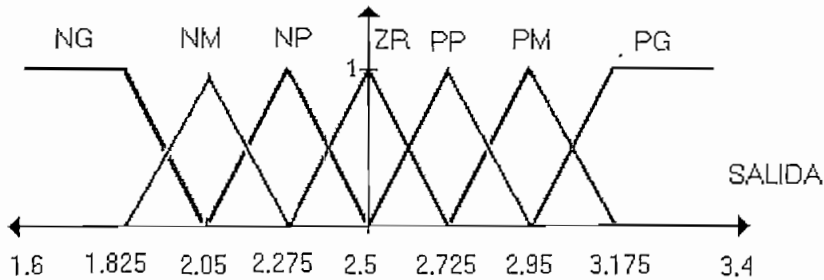


Figura 3.18. Fusificación de la salida.

Con fines prácticos se procede a normalizar los universos del discurso, esto se consigue por un simple escalonamiento.

Así para normalizar el universo de discurso del error, se escalona como se ilustra la figura 3.19.

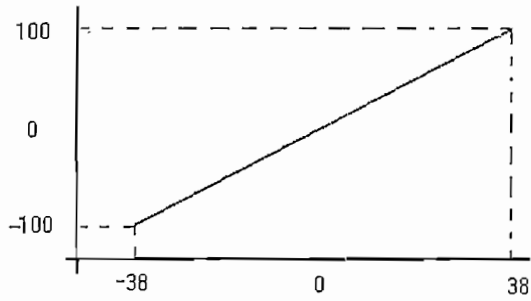


Figura 3.19. Escalamiento del universo del discurso del error

Para normalizar el universo de discurso de la tasa de cambio del error, se escala como se muestra en la figura 3.20

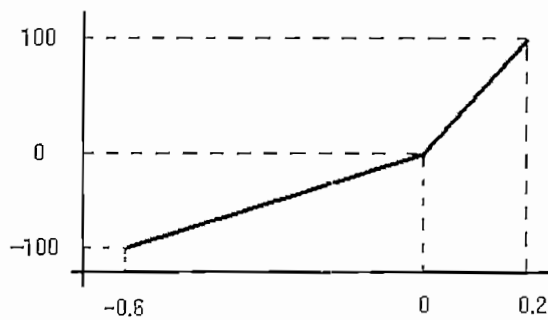


Figura 3.20. Escalamiento del universo de discurso de la tasa de cambio del error.

Para normalizar el universo de discurso de la salida (voltaje de salida al circuito de control), se escala como se muestra en la figura 3.21.

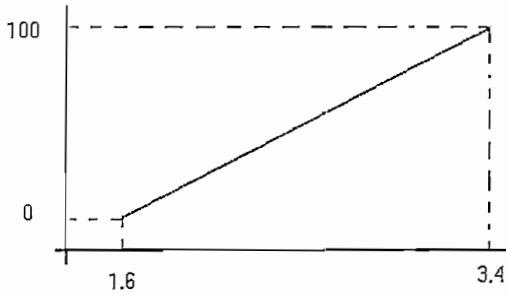


Figura 3.21. Escalamiento del universo del discurso de la salida.

Por lo tanto, los rangos normalizados de los universos de discurso del error, tasa de cambio del error y de la salida, son:

Error	[-100	100]
Δ Error	[-100	100]
Salida	[0	100]

Consecuentemente las figuras 3.16, 3.17 y 3.18 se ven modificadas en sus universos de discurso, como a continuación se muestra.

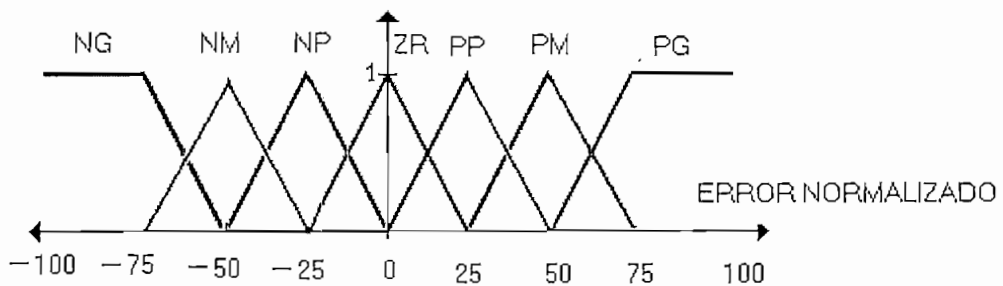


Figura 3.22. Normalización del universo del discurso del error.

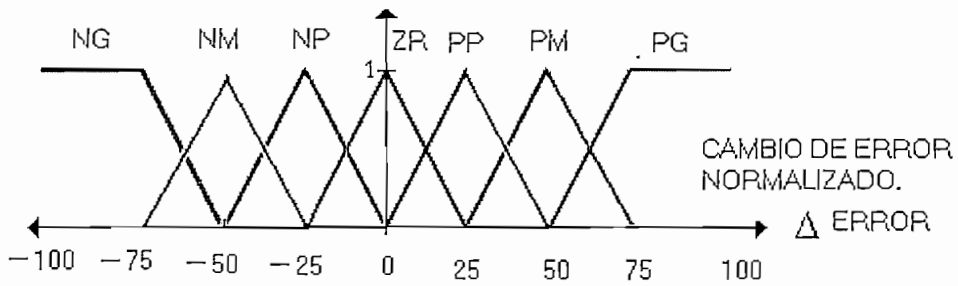


Figura 3.23. Normalización del universo del discurso de la tasa de cambio.

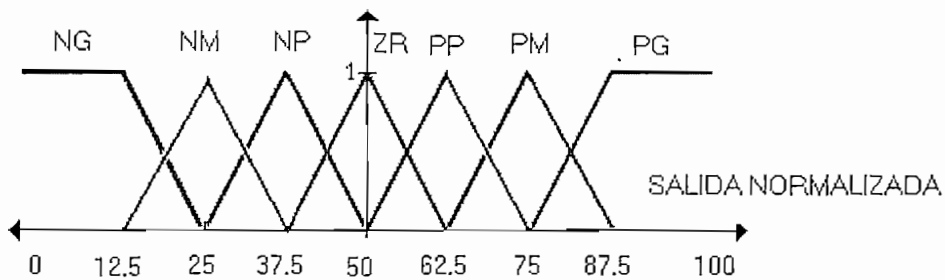


Figura 3.24. Normalización del universo del discurso de la salida .

Con esto se realiza el mecanismo de inferencia con la DLL.

3.2.4 INTERFACES GRAFICAS Y DIAGRAMAS DE FLUJO

Las interfaces gráficas que se indican en las figuras 3.25 y 3.26 muestran las pantallas que aparece cuando se ejecuta el programa de control.

La figura 3.25 tiene diferentes parámetros que se deben calibrar para poder realizar el control, a continuación se detalla cada uno de estos parámetros y las funciones que realizan.

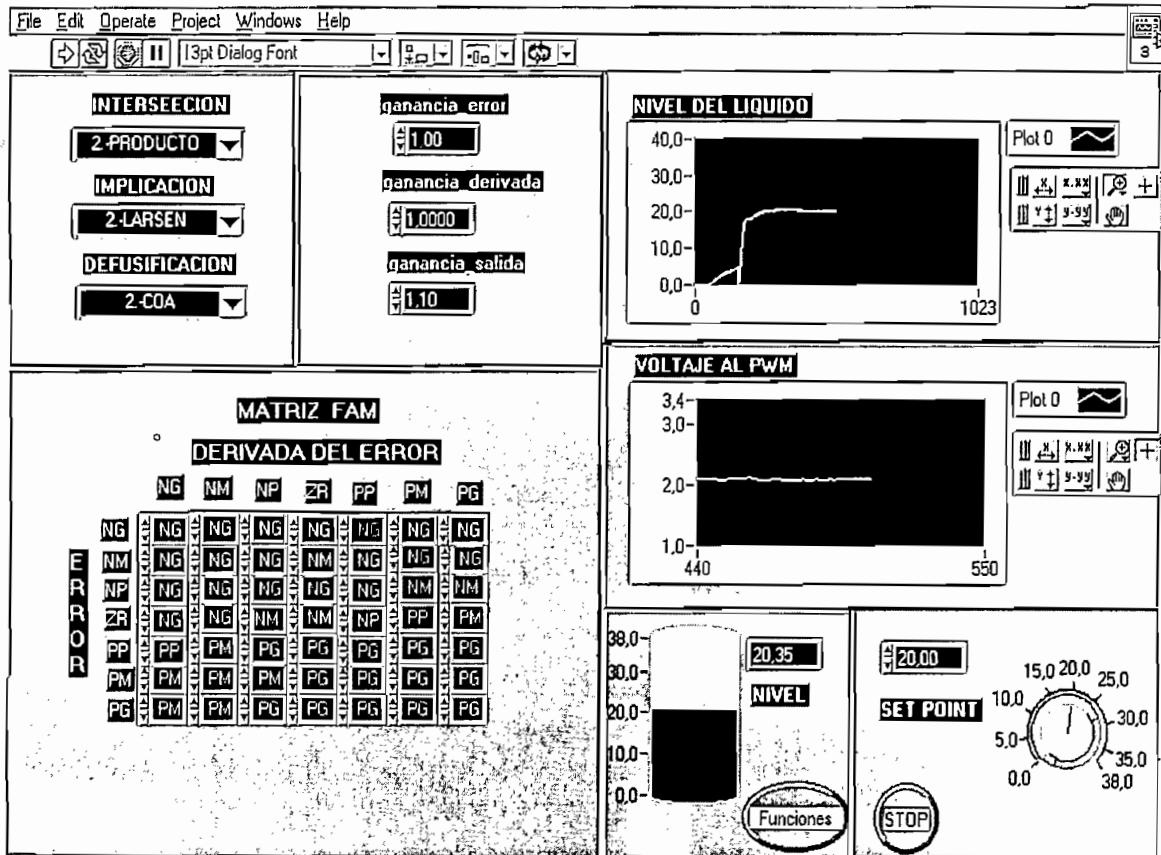


Figura 3.25. Interfaz gráfica de control.

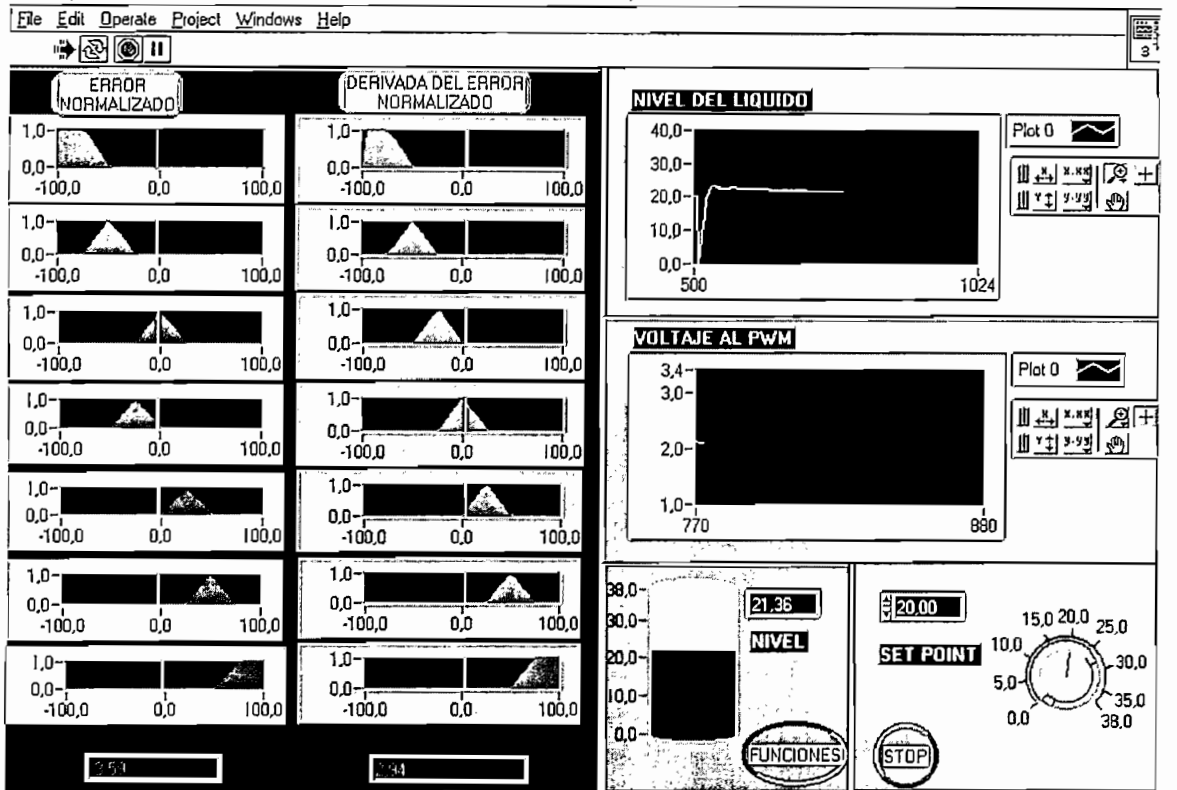


Figura 3.26. Interfaz gráfica de control cuando se presiona el botón funciones.

INTERSECCION .- Corresponde nos a la opción de seleccionar que tipo de intersección se desea, producto o mínimo.

Es preciso recordar que la intersección es aquella operación lógica se emplea cuando se activan más de una regla.

IMPLICACION .- Se elige el tipo de implicación que se desee, entre implicación de Mandani (mínimo) o la implicación de de Larsen (producto).

DEFUSIFICACION .- Se encuentran dos métodos de defusificación ,ya explicados por centro promedio (COA), o por centro de gravedad (COG) .

GANANCIA DE ERROR .- Lo que se realiza con este factor es expandir o contraer el universo del discurso del error ,para conseguir pasos de ajuste muy finos, se realiza esto si en caso se necesita para calibrar un poco mejor.

GANANCIA TASA DE CAMBIO DEL ERROR .- Lo que se realiza con este factor es expandir o contraer el universo del discurso de la velocidad de cambio del error , en pasos finos, como en el caso anterior.

GANANCIA DE SALIDA .- Lo que se realiza con este factor es expandir o contraer el universo del discurso de la salida , esta es otra buena opción para mejorar la sensibilidad del control.

MATRIZ FAM .- Esta matriz es la que almacena la base de conocimiento o la experiencia del operador, es muy flexible, pues es fácil de calibrar, con sólo elegir entre las siete diferentes opciones que se ha dado para escoger :

<i>NG</i>	Negativo Grande.
<i>NM</i>	Negativo Mediano.
<i>NP</i>	Negativo Pequeño.
<i>ZR</i>	Zero
<i>PP</i>	Positivo Pequeño.
<i>PM</i>	Positivo Mediano.
<i>PG</i>	Positivo Grande.

Para tal selección se debe tener un buen criterio como funciona la planta, y sobre todo la calibración se la realiza observando como se va comportando el sistema, hasta llegar a conseguir resultados satisfactorios.

NIVEL .-Este es un indicador numérico que indica el nivel del agua del tanque (mediante el sensor de presión diferencial) .

SET POINT .- Es la referencia a la que se desea que el control actúe.

INDICADORES GRAFICOS DE NIVEL DE LIQUIDO .- Representan en forma gráfica la señal de altura versus tiempo.

INDICADOR GRAFICO DE VOLTAJE .- Representan en forma grafica la señal de voltaje versus tiempo.

FUNCIONES .- Este botón sirve, para mostrar las funciones de pertenencia, tanto del error, como el cambio del error; aquí se puede ver como van cambiando el error y tasa de cambio hasta llegar a cero(en este momento la salida se ajusta a la referencia), además se tienen indicadores numéricos del error y tasa de cambio del error.

ERROR .- Este es un indicador de tipo numérico, que aparece en la figura 3.26 para ver el comportamiento de la acción del control.

TASA DE CAMBIO DEL ERROR.- Al igual que el anterior, es un tipo de indicador numérico, que aparece para observar como va cambiando esta variable a medida que se realiza la acción de control.

STOP.- Este botón sirve para detener la programa.

A continuación los diagramas de flujo de los dos programas realizados tanto del realizado en el Labview, como del Turbo C++.

Diagrama de flujo del programa en Labview.

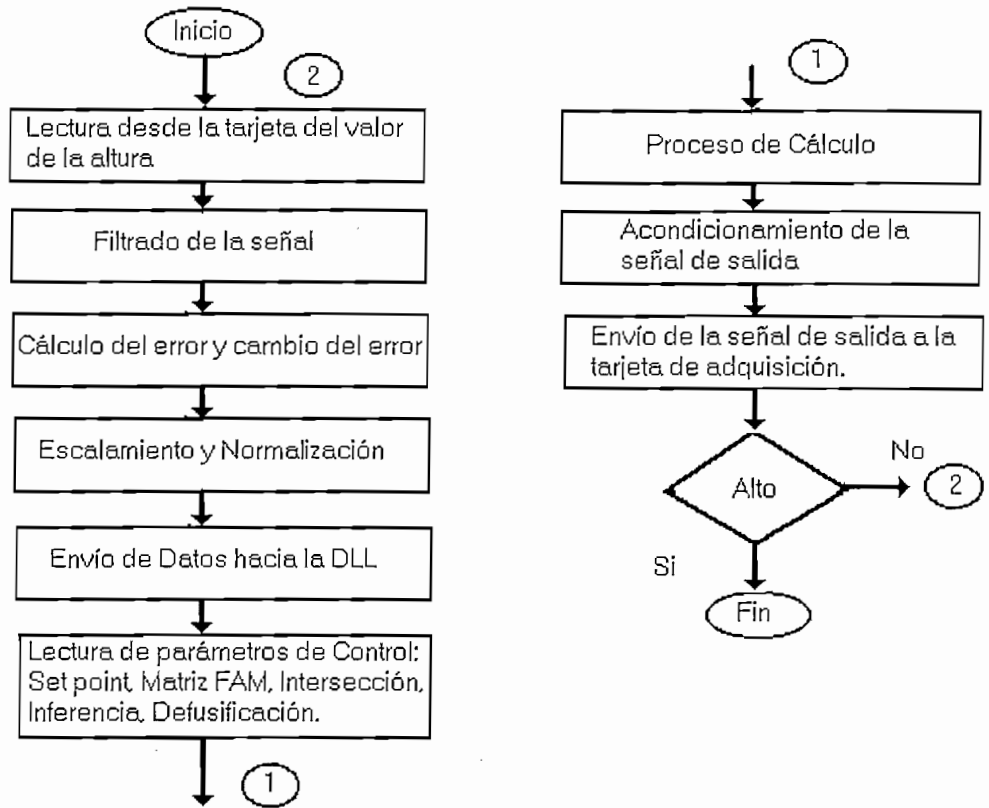
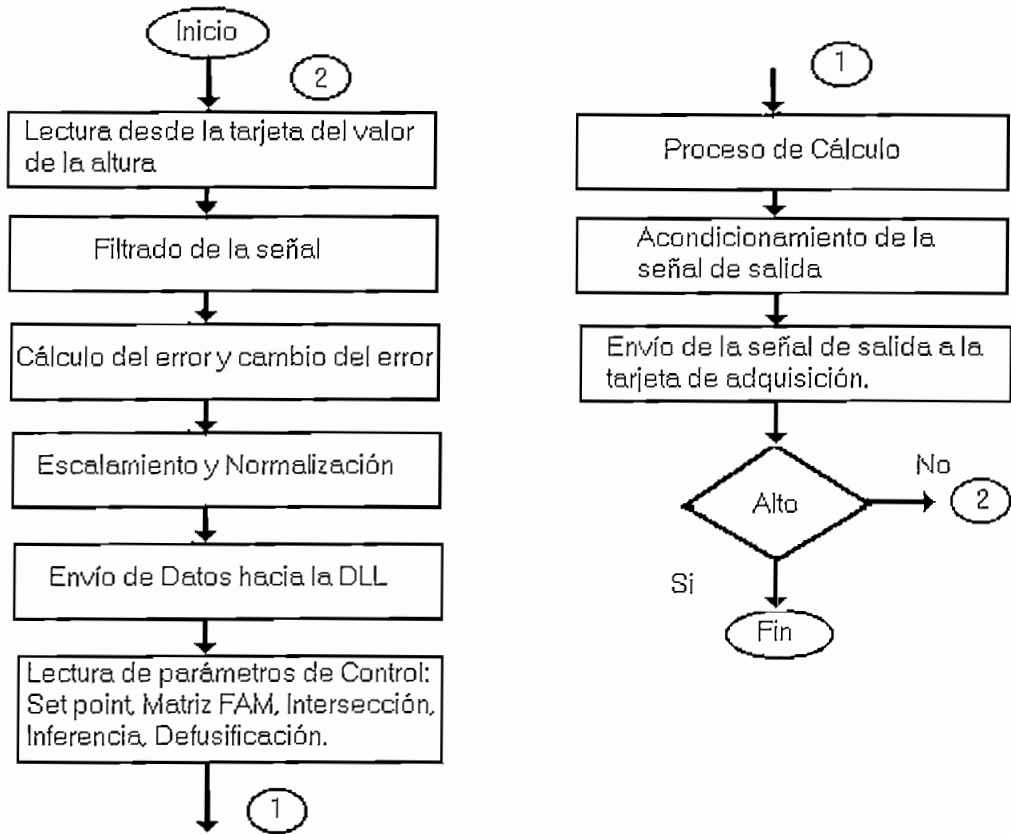


Diagrama de flujo del programa en turbo C++



Respecto al detalle de la programación, se adjuntan en los anexos los códigos de los programas tanto para Labview, como para Turbo C++.

CAPITULO 4

RESULTADOS EXPERIMENTALES

El conjunto de resultados que se muestran a continuación han sido seleccionados entre los múltiples obtenidos en las fases experimentales, a fin de demostrar la validez del presente trabajo.

Los gráficos que se presentan corresponden, los de el lado izquierdo a la señal de la respuesta del sistema, y los de la derecha al voltaje de control .

Como se tienen tres parámetros a seleccionar (intersección, implicación, defusificación) y cada uno de estos parámetros poseen a su vez dos opciones, se tendrán una combinación de ocho pruebas que realizar, que a continuación se muestran.

Para realizar un análisis de los resultados obtenidos, y mirar su comportamiento, se ha procedido a medir tres parámetros de respuesta que son:

Sobre-impulso máximo ($Mp\%$) .- Este parámetro indica el valor pico máximo de la curva de respuesta medido desde la unidad. Si el valor final estabilizado de la respuesta difiere de la unidad, se suele utilizar el sobreimpulso porcentual máximo (este es el que se determinó).

Tiempo de establecimiento(Ts) .-Es el que la curva de respuesta requiere para alcanzar y mantenerse en un rango alrededor del valor final .

Error de posición (Ep).- Este es un error en estado estable expresado en porcentaje.

4.1 CONJUNTO DE PRUEBAS I

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Producto

Implicación : Producto de Mandani

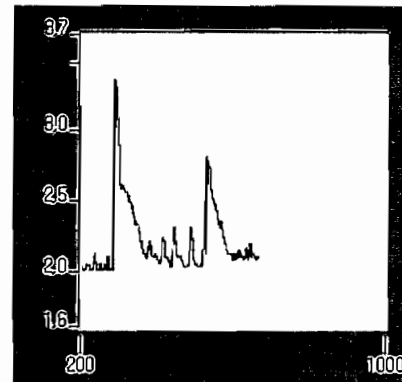
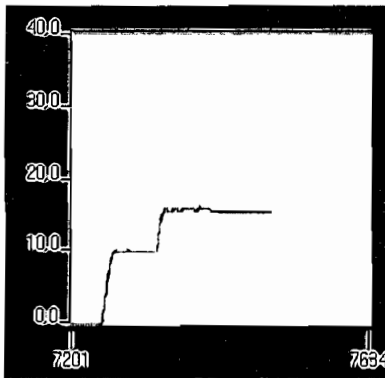
Defusificación : Centro de Gravedad (COG)

Válvula de desfoge completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

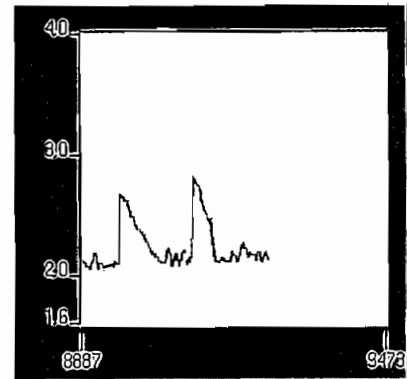
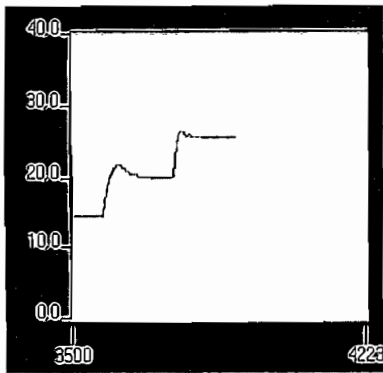
PRUEBA A



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 10cm	0	76	3
De 10cm a 15cm	10	50	2

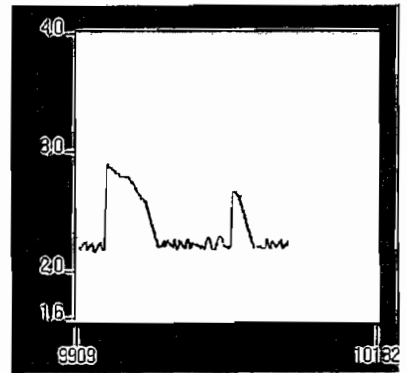
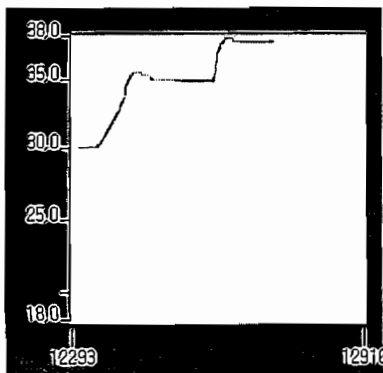
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 15 cm a 20cm	35	38	2
De 20cm a 25cm	10	40	4

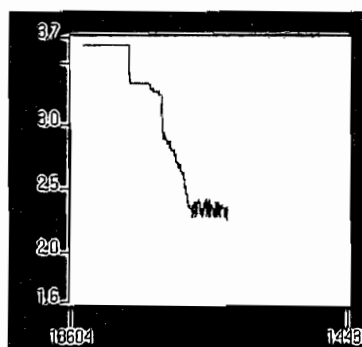
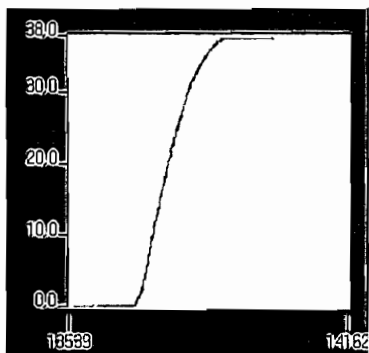
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 30 cm a 35cm	20	48	2
De 35cm a 38cm	10	49	3,33

PRUEBA D



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 38cm	0	160	0.52

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

4.2 CONJUNTO DE PRUEBAS II

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Producto

Implicación : Mínimo de Mandani

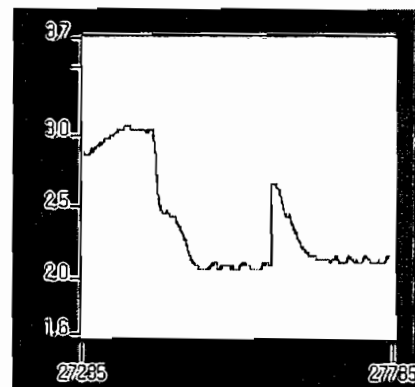
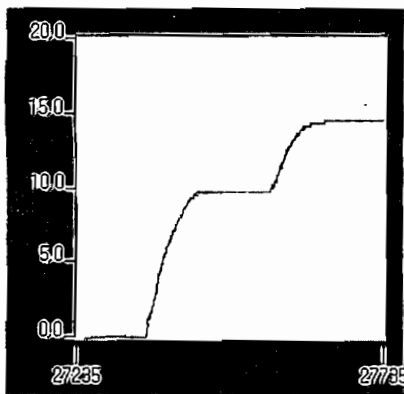
Defusificación : Centro de Gravedad (COG)

Válvula de desfogo completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

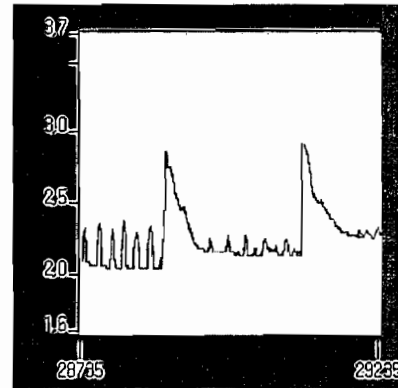
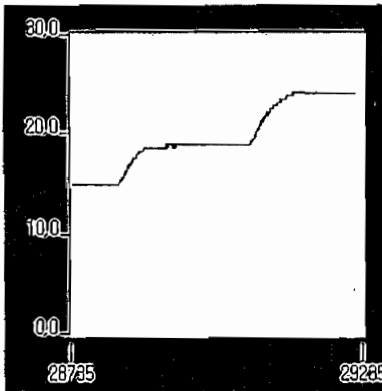
PRUEBA A:



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 10cm	0	45	2
De 10cm a 15cm	0	48	4

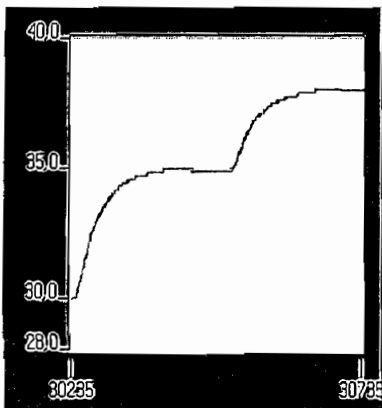
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 15 cm a 20cm	0	45	10
De 20cm a 25cm	0	36	4

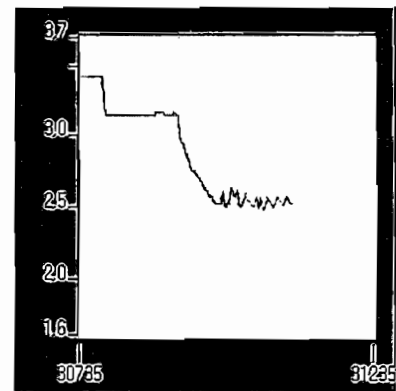
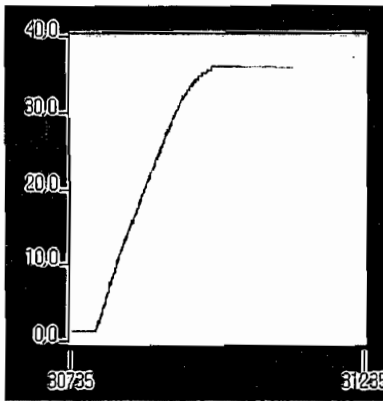
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 30 cm a 35cm	0	48	2
De 35cm a 38cm	0	40	6,66

PRUEBA D



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 38cm	0	100	0.7

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

4.3 CONJUNTO DE PRUEBAS III

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Mínimo

Implicación : Producto de Larsen.

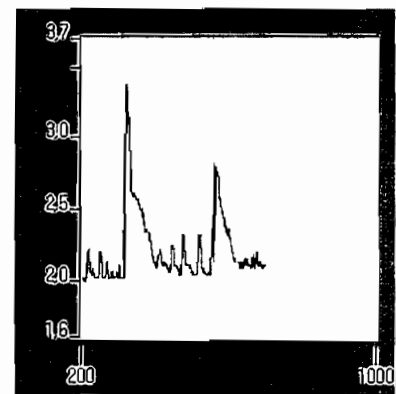
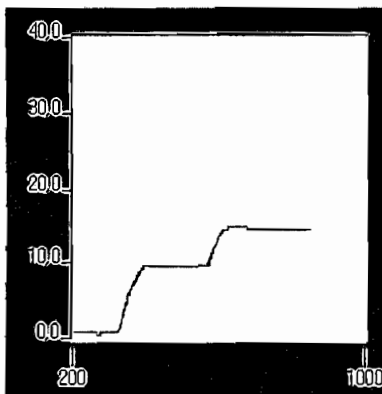
Defusificación : Centro de Gravedad (COG)

Válvula de desfoge completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

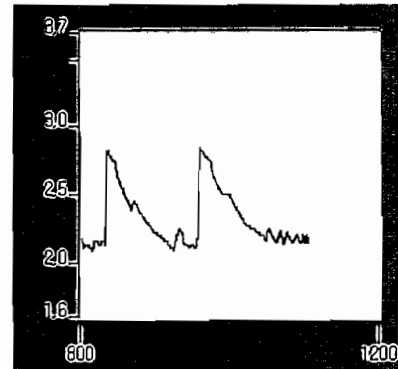
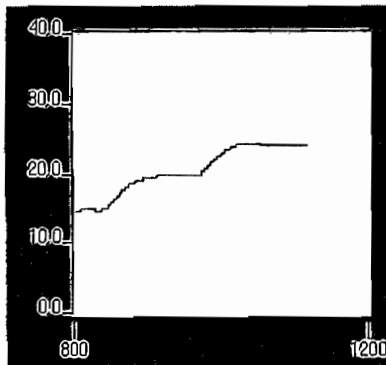
PRUEBA A



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 10cm	0	50	1
De 10cm a 15cm	10	30	2

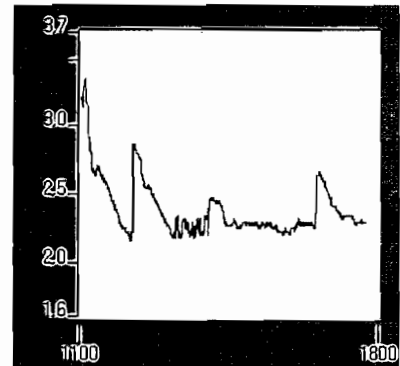
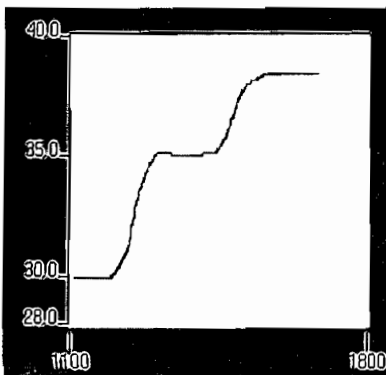
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts (Seg)	Ep(%)
De 15 cm a 20cm	0	35	0
De 20cm a 25cm	0	37	2

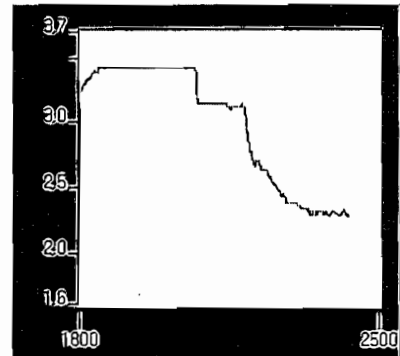
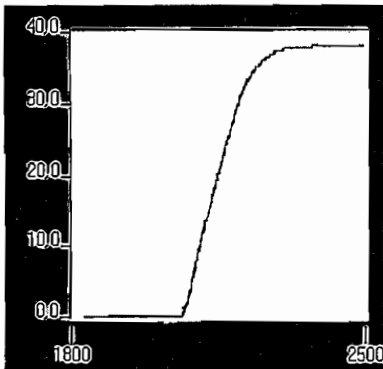
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 30 cm a 35cm	10	30	2
De 35cm a 38cm	0	35	10

PRUEBA D.



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0cm a 38 cm	0	120	0.78

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

4.4 CONJUNTO DE PRUEBAS IV

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Producto

Implicación : Producto de Larsen.

Defusificación : Centro de Gravedad (COG)

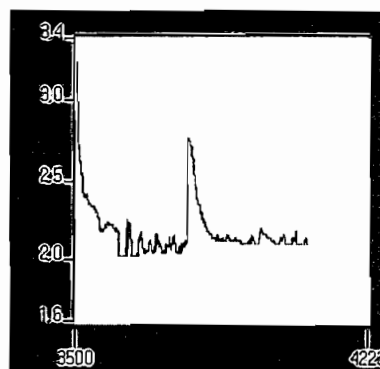
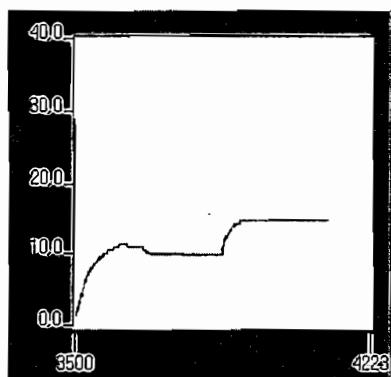
Ganancia de salida 1.1

Válvula de desfoge completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

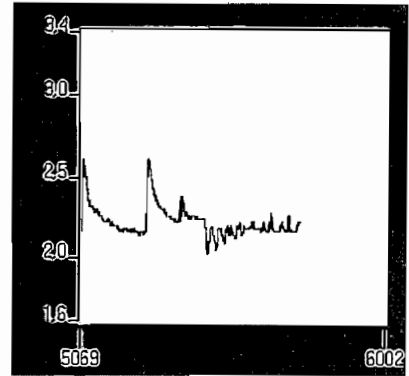
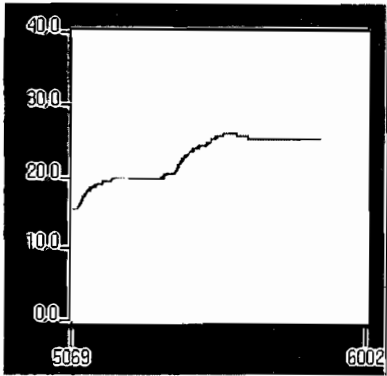
PRUEBA A



Con esta prueba se obtienen los siguientes valores:

PERTURBACION	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 10cm	25	40	1
De 10cm a 15cm	0	37	8

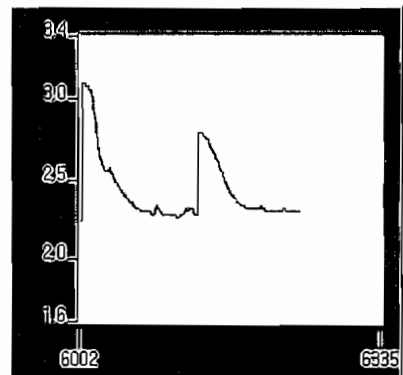
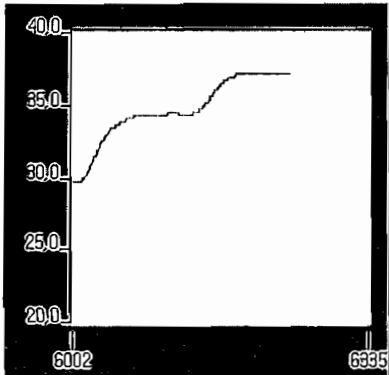
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg.)	Ep(%)
De 15 cm a 20cm	0	50	2
De 20cm a 25cm	20	55	8

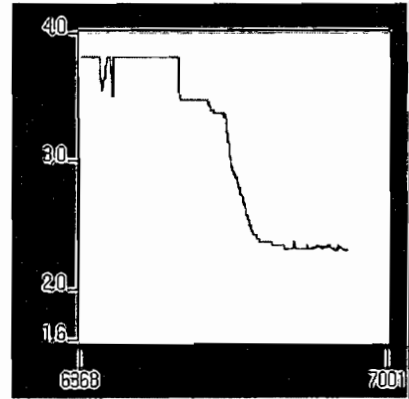
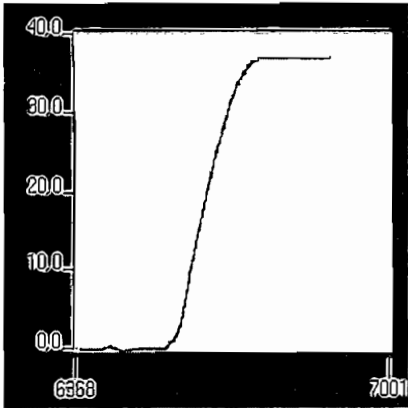
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 30 cm a 35cm	0	41	4
De 35cm a 38cm	0	37	10

PRUEBA D



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 38cm	0	40	0.78

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

4.5 CONJUNTO DE PRUEBAS V

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Mínimo

Implicación : Mínimo de Mandani.

Defusificación : Centro Promedio (COA)

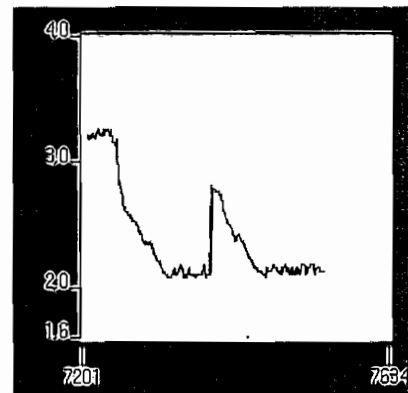
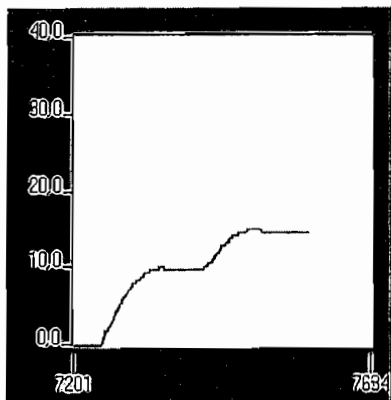
Ganancia de salida 1.0

Válvula de desfoge completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

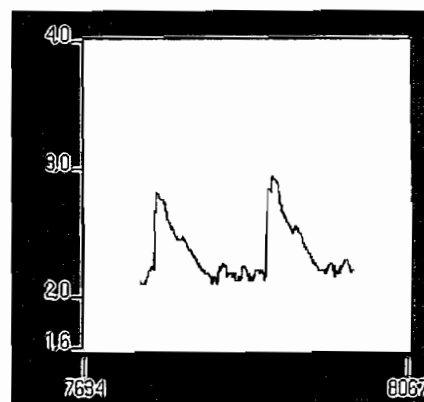
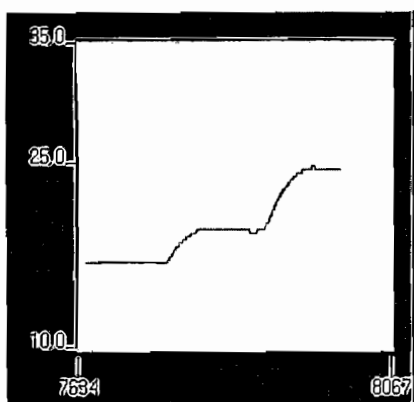
PRUEBA A



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 10cm	8	37	1
De 10cm a 15cm	10	29	2

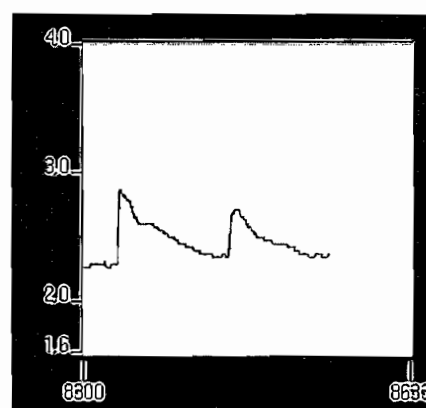
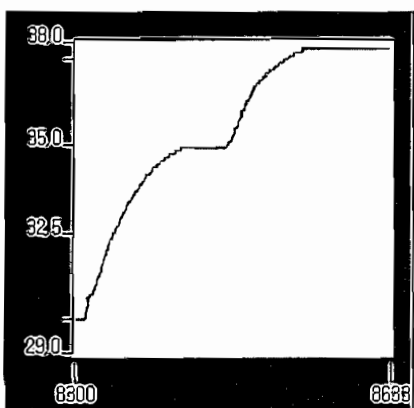
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 15 cm a 20cm	0	40	6
De 20cm a 25cm	2	41	4

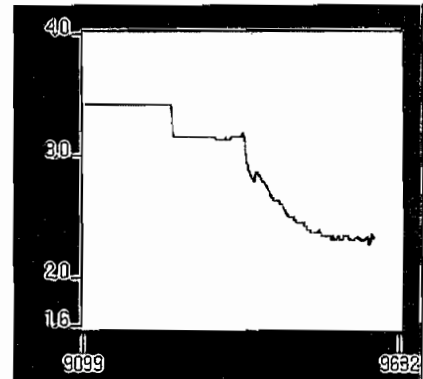
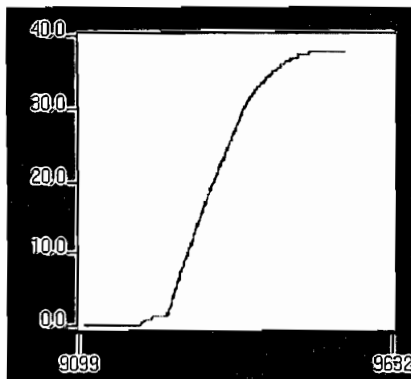
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 30 cm a 35cm	0	39	2
De 35cm a 38cm	0	35	3,33

PRUEBA D



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	$M_p(\%)$	$T_s(\text{Seg})$	$E_p(\%)$
De 0 cm a 38cm	0	120	0.26

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

4.6 CONJUNTO DE PRUEBAS VI

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Producto

Implicación : Mínimo de Mandani.

Defusificación : Centro Promedio (COA)

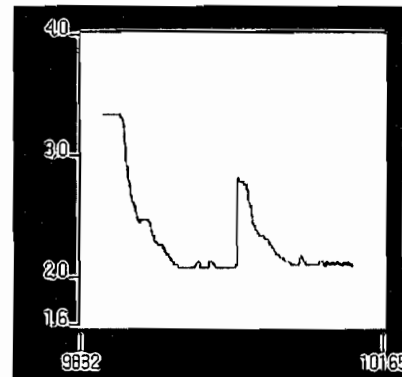
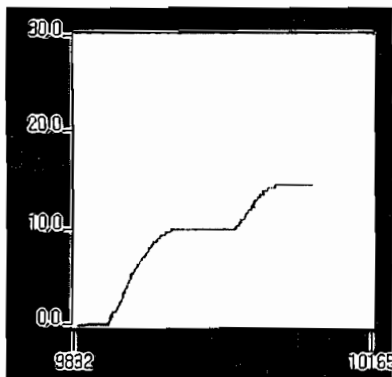
Ganancia de salida 1.0

Válvula de desfoge completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

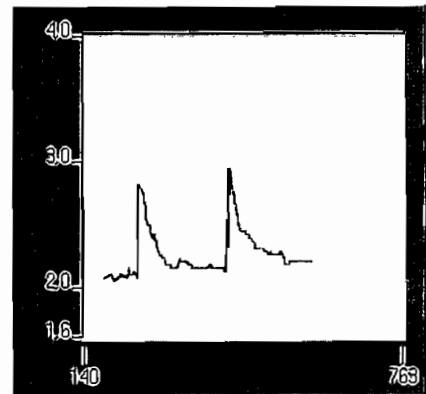
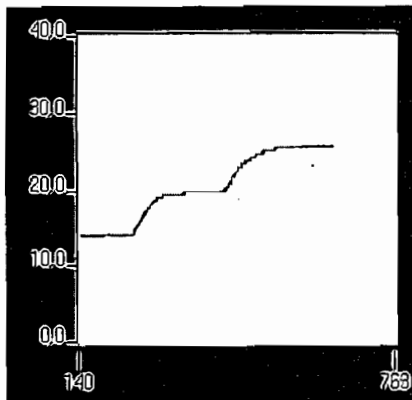
PRUEBA A



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 10cm	0	37	3
De 10cm a 15cm	0	38	4

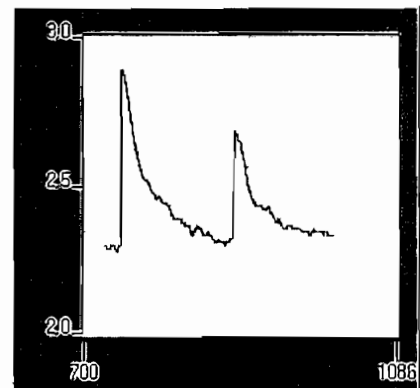
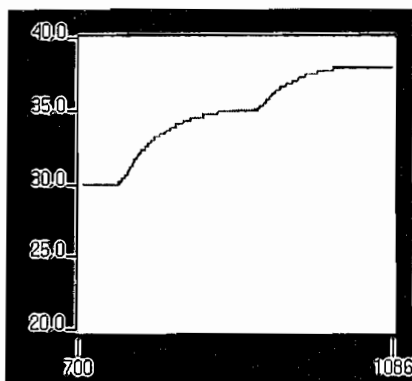
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 15 cm a 20cm	0	40	4
De 20cm a 25cm	0	37	4

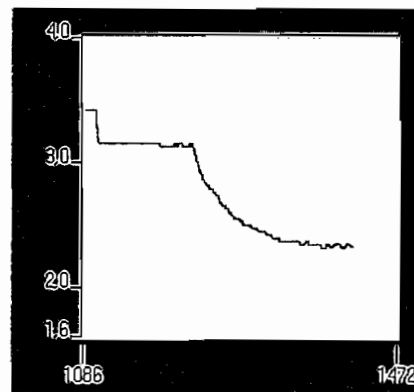
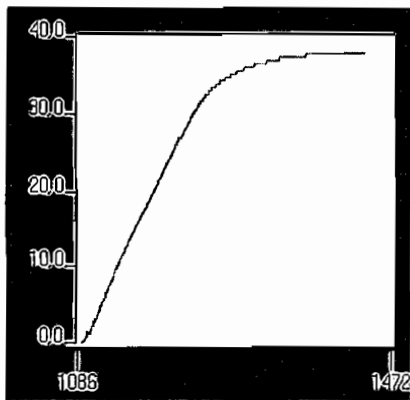
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 30 cm a 35cm	0	50	2
De 35cm a 38cm	0	38	3,33

PRUEBA D



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 38cm	0	138	0.26

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

4.7 CONJUNTO DE PRUEBAS VII

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Producto

Implicación : Producto de Larsen.

Defusificación : Centro Promedio (COA)

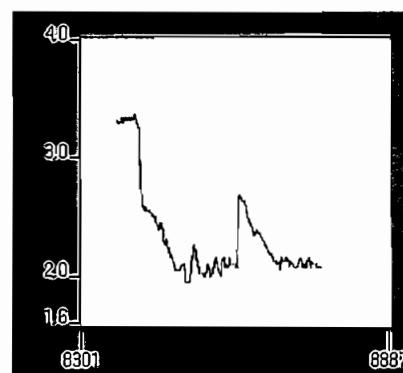
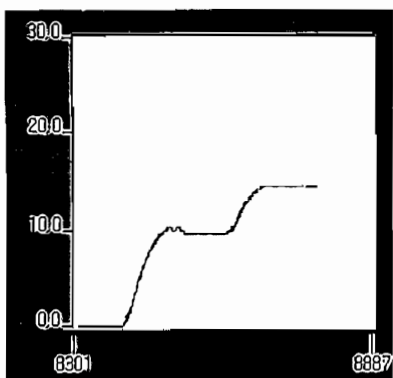
Ganancia de salida 1.0

Válvula de desfoge completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

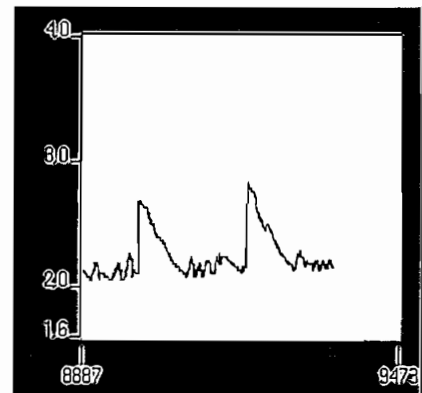
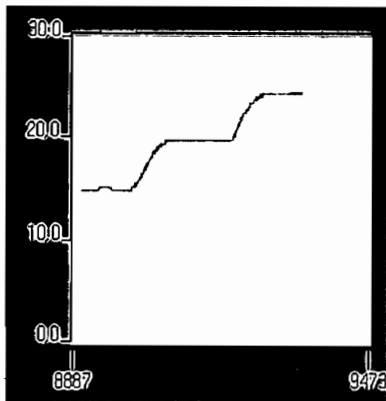
PRUEBA A



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 10cm	10	38	2
De 10cm a 15cm	0	40	6

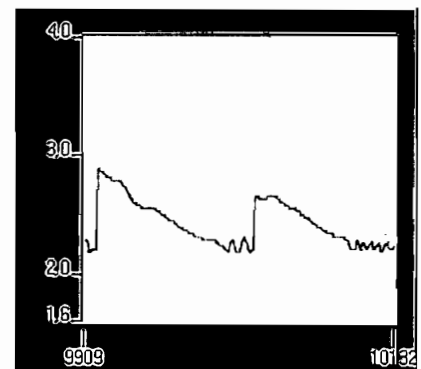
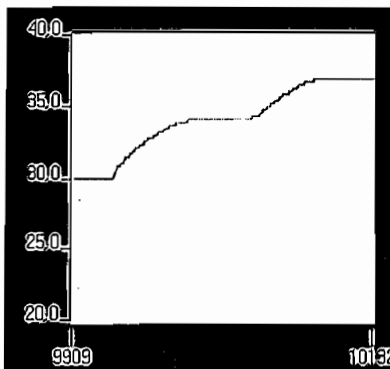
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 15 cm a 20cm	0	40	6
De 20cm a 25cm	0	45	2

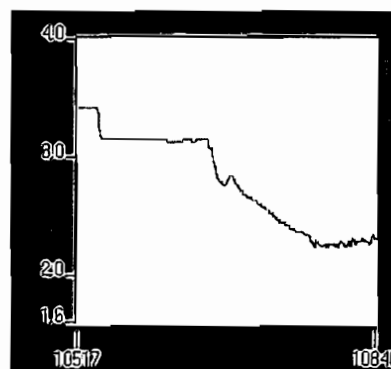
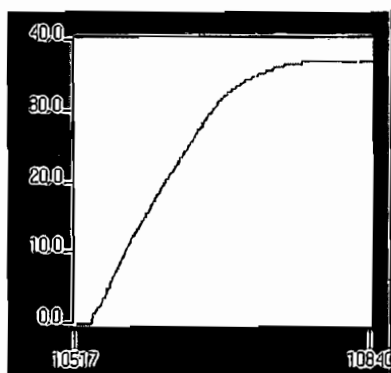
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 30 cm a 35cm	0	50	2
De 35cm a 38cm	0	45	10

PRUEBA D



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep(%)
De 0 cm a 38cm	0	155	0,52

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

4.8 CONJUNTO DE PRUEBAS VIII

Las condiciones en que se realizó esta prueba son las siguientes:

Intersección : Mínimo

Implicación : Producto de Larsen.

Defusificación : Centro Promedio (COA)

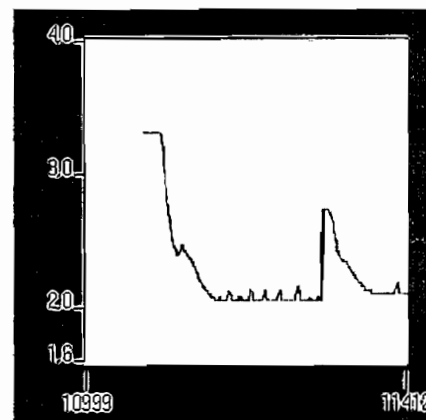
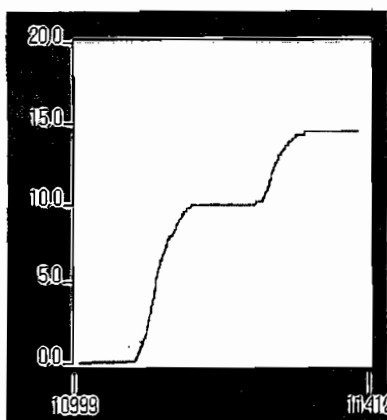
Ganancia de salida :1.0

Válvula de desfoge completamente abierta.

Se variaron las reglas de la matriz de inferencia hasta obtener resultados satisfactorios.

Las pruebas de perturbación realizadas una vez en equilibrio, bajo estas condiciones y sus respectivos resultados se muestran a continuación.

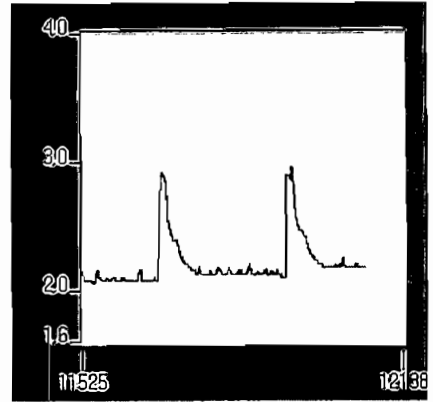
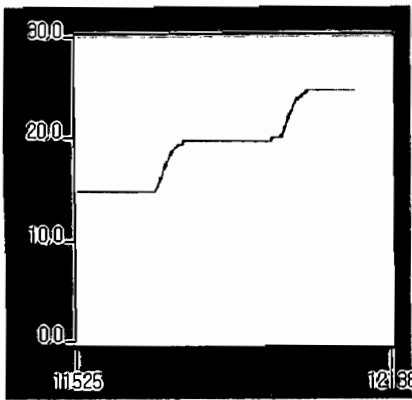
PRUEBA A



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep
De 0 cm a 10cm	0	32	1
De 10cm a 15cm	0	33	2

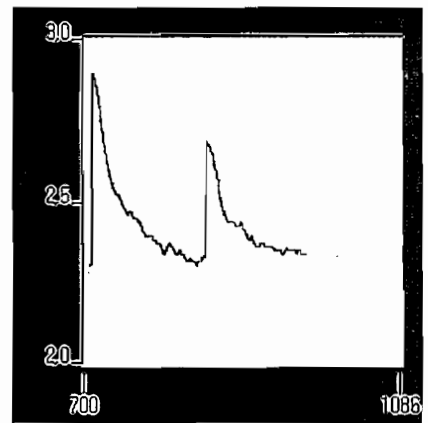
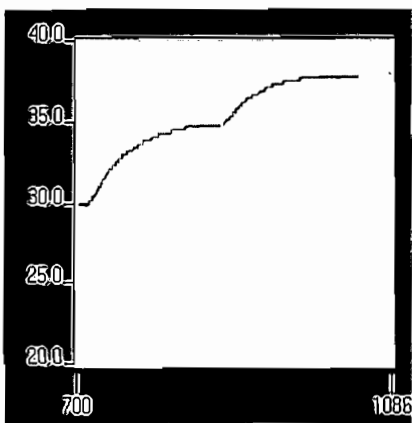
PRUEBA B



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep
De 15 cm a 20cm	0	37	2
De 20cm a 25cm	0	35	2

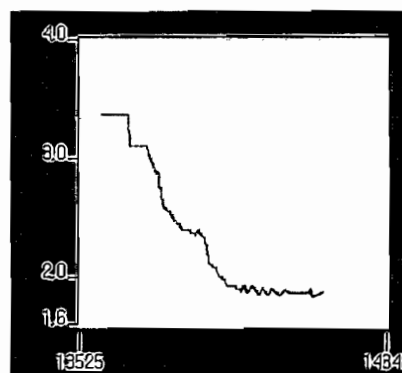
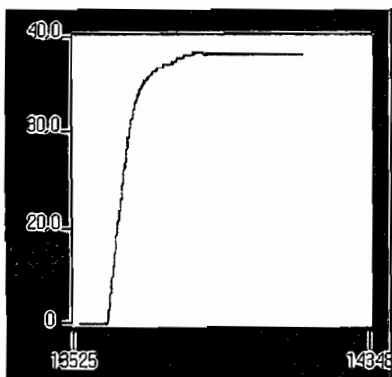
PRUEBA C



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep
De 30 cm a 35cm	0	35	2
De 35cm a 38cm	0	38	10

PRUEBA D



Con esta prueba se obtienen los siguientes valores:

CAMBIO DE SET POINT	Mp(%)	Ts(Seg)	Ep
De 0 cm a 38cm	0	135	0.26

De este conjunto de prueba realizadas, se puede ver que la acción por parte del controlador difuso, satisfacen los objetivos que se propusieron en este proyecto, según los resultados obtenidos.

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Las presentes conclusiones se citarán en orden de importancia, y de acuerdo a los pasos para el desarrollo de este proyecto.

La ventaja principal e importante del uso de un controlador difuso, es que no se necesita el modelo matemático de la planta, ya que este tipo de controlador, utiliza expresiones lingüísticas, en vez de formulas matemáticas.

Respecto al sensor cuando se utilizaba directamente la señal proveniente del circuito de acondicionamiento del sensor, se obtenían resultados de cambio del error que no tenían significado debido a la contaminación con ruido, razón por la cual no servía para la entrada de nuestro controlador; para superar este problema se utilizó un filtro digital.

Los parámetros de este filtro como orden, frecuencia de muestreo, frecuencia de corte, se los determinó experimentalmente. Hay que tomar en cuenta, que si elevamos el orden del filtro se incrementa el retardo en la respuesta del sensor, y además se produce una atenuación de la señal , perjudicando al control en tiempo real.

Respecto al control difuso los universos del discurso del error, cambio del error y salida se los determinó realizando pruebas experimentales, y una vez obtenidos se procedió a particionarlos con funciones de pertenencia, para luego normalizarlos, para trabajar en forma porcentual.

El escalonamiento de la señal, es decir la normalización del universo del discurso, es muy útil para tener una mejor percepción de las variables que intervienen en el control : error, cambio del error, salida o voltaje.

Respecto a los lenguajes utilizados, la función principal de la DLL es la de permitir la interacción entre el programa realizado en turbo C++ con el Labview,

El circuito de control y de potencia no tienen la misma referencia, pues se hallan aislados por medio de un opto-acoplador la razón principal para la utilización de este elemento es para la protección de la tarjeta de Labview.

En general este software desarrollado es muy versátil, puede ser utilizado también para realizar el control de otro sistema, pues lo único que cambian son los rangos del universo del discurso, y la calibración de la matriz de inferencia, siempre y cuando se utilicen las mismas funciones de pertenencia triangulares; claro está que para otras plantas o sistemas en las que la aplicación de estas funciones de pertenencia triangulares no logren un buen comportamiento del sistema a ser controlado ,se debería cambiar por otras funciones de pertenencia que mejor se ajusten a estas plantas.

Los resultados han sido en general satisfactorios, pero el caso en el que se obtuvieron los mejores resultados fueron para el conjunto de pruebas v (intersección mínimo, implicación de Mandani, defusificación COA) y para el conjunto de pruebas viii (intersección mínimo, implicación Larsen, defusificación COA), pues sus tiempos de establecimiento, máximo nivel sobreporcentual, y errores de posición fueron mejores al resto de pruebas.

En general todas las pruebas, muestran un adecuado comportamiento del control, observándose que es muy fácilmente calibrar, los parámetros para llegar a obtener resultados eficientes.

Por último se cree haber cumplido a cabalidad todos los objetivos, y que el desarrollo del presente trabajo a sido una tarea gratificadora y enriquecedora

5.2 RECOMENDACIONES

El filtrado de la señal proveniente del sensor es una buena opción, para sensores que no posean buenas características con respecto al ruido, de esta forma que puedan servir para que funcionen adecuadamente en los controladores difusos. Teniendo siempre presente no utilizar filtros de ordenes muy altos, ya que se verá su efecto en una atenuación y retardo de la señal.

Cuando se vaya a utilizar este tipo de controlador, se deberá elegir al sensor, con características que lo hagan menos sensible al ruido; sin embargo, mientras mejor es el sensor su costo se eleva.

Para el diseño de controladores difusos, se debería empezar a particionar con funciones de pertenencia triangulares, ya que con las otras funciones es más difícil sacar el grado de pertenencia, y por cuestiones de facilidad en el cálculo. Pero si el control no es tan bueno, se deben cambiar por otras funciones de pertenencia.

Cuando se realice el control en tiempo real por medio del computador, se deben recurrir a programas con tiempos de procesamiento muy rápidos, para llegar a realizar un control eficiente.

En el caso que se utilice Labview, se tiene necesariamente que construir DLLs para que el proceso de ejecución sea lo menor posible.

Cuando en la implementación se utilicen por un lado circuitos de control, y por otros los circuitos de potencia siempre es bueno aislarlos a ambos, para que un daño que se produzca no afecte a ambos.

Se debe tomar en cuenta que la bomba de agua no debe trabajar en vacío, es decir sin líquido, pues se vería afectada su vida útil.

Cuando se utilice el elemento LM3524, es bueno tratar de utilizar el terminal del "shut down", este permite que se interrumpa la generación del PWM para casos

de emergencia, en esta aplicación la bomba dejará de funcionar; pues éste terminal está conectado a un sensor final de carrera tipo boya, que nos indica cuando el agua ha alcanzado la altura máxima, con lo que se evita que se desborde.

Este tipo de controladores son muy útiles cuando es una tarea compleja y tediosa obtener el modelo matemático de la planta, de aquí que se recomienda esta tecnología para ser usada, pues este tipo de controlador funciona con un modelo cualitativo de la planta .

BIBLIOGRAFIA

- [1] Chuen Chien Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I" *IEEE Transactions on Systems, Man and Cybernetics*, Vol.20, March/April 1990.
- [2] Chuen Chien Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II", *IEEE Transactions on Systems, Man and Cybernetics*, Vol.20, March/April 1990.
- [3] Jan Jantzen , "Tutorial on Fuzzy Logic", Department of Automation-*Technical University of Denmark*, August 1998.
- [4] Jan Jantzen , "Design of Fuzzy Controllers", Department of Automation-*Technical University of Denmark*, August 1998.
- [5] R.Langari, R Berenji, "Fuzzy Logic in Control Engineering", *Department of Mechanical Engineering, Texas A &M. University*.
- [6] L.A.Bryan,E.A.Bryan,"Programable Controllers,Theory and Implementation",*Industrial Text Company*,2nd edition,Chapter Seventeen,1997.
- [7] F.Gomide, R Gudwin, A.Netto, "Controle de Processos por Logica Fuzzy".
- [8] A.Ibrahim, "Introduction to Applied Fuzzy Electronics", Prentice Hall, New Jersey ,1997.
- [9] J. M. Mendel, "Fuzzy Logic Systems for Engineering: A Tutorial", *Proceedings of the IEEE*, March 1995.

[10] M. Johansson, "A. Primer on Fuzzy Control", *Department of Automatic Control-Lund Institute of Technology, 1996.*

ANEXOS

ANEXO A MANUAL DE USUARIO

A.1 DESCRIPCION GENERAL DEL PROGRAMA

Este es un programa que nos permite realizar el control difuso de un nivel de líquido en tiempo real, en el que dispone de algunos parámetros para su calibración, y de la buena elección de estos dependerá el buen funcionamiento del sistema.

A.2 REQUERIMIENTOS DEL SISTEMA

Para utilizar el programa adecuadamente se deben tener los siguientes requerimientos mínimos.

- Procesador Pentium de 133 MHz.
- 8MB de memoria RAM.
- Labview versión 5.0 (incluido el software correspondiente a adquisición de datos).
- Tarjeta de adquisición de datos Lab-PC-1200.
- Tamaño del programa: 20 MB de disco duro.
- Windows 95.

A.3 CONEXIONES DE HARDWARE NECESARIAS

Se deben seguir los siguientes pasos:

Conectar la señal proveniente del sensor a un a entrada analógica de la tarjeta de adquisición de datos.

Alimentar al circuito de potencia y control con fuentes de voltaje, dos fuentes de voltaje independientes para el circuito de control, y otra fuente para que alimente al circuito de potencia.

Conectar la salida de la tarjeta de adquisición de datos , a la entrada del circuito de control.

Conectar la salida del circuito de potencia, a la bomba de agua.

Conectar también el final de carrera , a la entrada del circuito de control para casos de emergencia.

A.4 UTILIZACION DEL PROGRAMA

Para poder utilizar este programa será necesario tener en el disco duro de la computadora la carpeta de programas en los que están los archivos con extensión vi, y la DLL.

La figura 3.25 tiene diferentes parámetros que se deben calibrar para poder realizar el control, a continuación se detalla cada uno de estos parámetros y las funciones que realizan.

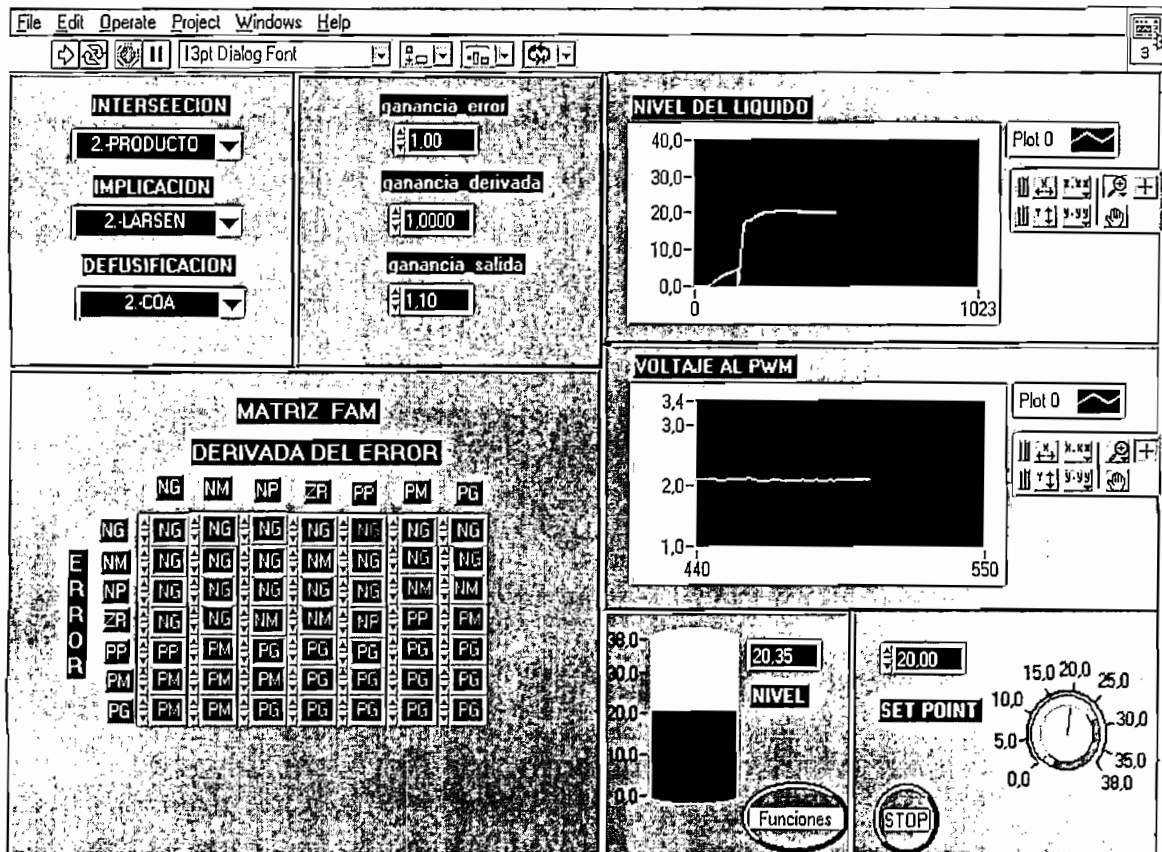


Figura 3.25. Interfaz gráfica de control.

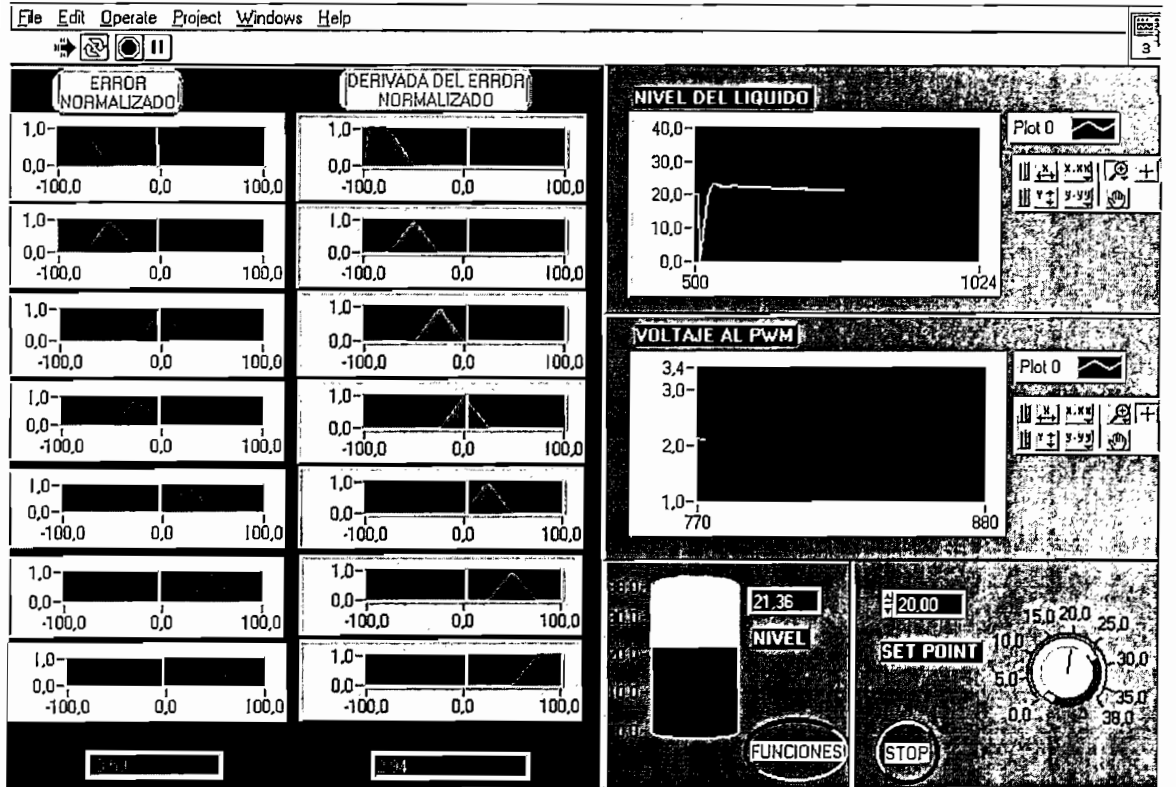


Figura 3.26. Interfaz gráfica de control cuando se presiona el botón funciones.

INTERSECCION .- Corresponde nos a la opción de seleccionar que tipo de intersección se desea, producto o mínimo.

Es preciso recordar que la intersección es aquella operación lógica se emplea cuando se activan más de una regla.

IMPLICACION .- Se elige el tipo de implicación que se desee, entre implicación de Mandani (mínimo) o la implicación de de Larsen (producto).

DEFUSIFICACION .- Se encuentran dos métodos de defusificación ,ya explicados por centro promedio (COA), o por centro de gravedad (COG) .

GANANCIA DE ERROR .- Lo que se realiza con este factor es expandir o contraer el universo del discurso del error ,para conseguir pasos de ajuste muy finos, se realiza esto si en caso se necesita para calibrar un poco mejor.

GANANCIA TASA DE CAMBIO DEL ERROR .- Lo que se realiza con este factor es expandir o contraer el universo del discurso de la velocidad de cambio del error , en pasos finos, como en el caso anterior.

GANANCIA DE SALIDA .- Lo que se realiza con este factor es expandir o contraer el universo del discurso del la salida , esta es otra buena opción para mejorar la sensibilidad del control.

MATRIZ FAM .- Esta matriz es la que almacena la base de conocimiento o la experiencia del operador, es muy flexible, pues es fácil de calibrar, con sólo elegir entre las siete diferentes opciones que se ha dado para escoger :

<i>NG</i>	Negativo Grande.
<i>NM</i>	Negativo Mediano.
<i>NP</i>	Negativo Pequeño.
<i>ZR</i>	Zero
<i>PP</i>	Positivo Pequeño.
<i>PM</i>	Positivo Mediano.
<i>PG</i>	Positivo Grande.

Para tal selección se debe tener un buen criterio como funciona la planta, y sobre todo la calibración se la realiza observando como se va comportando el sistema, hasta llegar a conseguir resultados satisfactorios.

NIVEL .-Este es un indicador numérico que indica el nivel del agua del tanque (mediante el sensor de presión diferencial) .

SET POINT .- Es la referencia a la que se desea que el control actúe.

INDICADORES GRAFICOS DE NIVEL DE LIQUIDO .- Representan en forma gráfica la señal de altura versus tiempo.

INDICADOR GRAFICO DE VOLTAJE .- Representan en forma grafica la señal de voltaje versus tiempo.

FUNCIONES .- Este botón sirve, para mostrar las funciones de pertenencia, tanto del error, como el cambio del error; aquí se puede ver como van cambiando el error y tasa de cambio hasta llegar a cero(en este momento la salida se ajusta a la referencia), además se tienen indicadores numéricos del error y tasa de cambio del error.

ERROR .- Este es un indicador de tipo numérico, que aparece en la figura 3.26 para ver el comportamiento de la acción del control.

TASA DE CAMBIO DEL ERROR.- Al igual que el anterior, es un tipo de indicador numérico, que aparece para observar como va cambiando esta variable a medida que se realiza la acción de control.

STOP.- Este botón sirve para detener la programa.

ANEXO B LISTADO DE PROGRAMAS

B.1 PROGRAMA EN TURBO C++

A continuación el programa escrito en TURBO C++

```

//*****ESCUELA POLITECNICA NACIONAL.*****
//*****FACULTAD DE INGENIERIA ELECTRICA.*****
//*****DEPARTAMENTO.*****
CONTROL Y AUTOMATIZACION INDUSTRIAL.*****
//*****TESIS PREVIA A LA OBTENCION DEL TITULO*****
//*****PROGRAMA:CONTROL DIFUSO DE UN NIVEL DE LIQUIDO EN
TIEMPO REAL.
//*****REALIZAD POR:
//*****FALCONI TELLO NELSON FERNANDO.*****
//*****DIRECTOR DE TESIS:ING.MARCO BARRAGAN.*****

//DECLARACION DE ESTRUCTURA PARA ALMACENAR LOS DATOS DE:
//LOS CONJUNTOS DIFUSOS DEL ERROR,CAMBIO DEL ERROR Y SALIDA.

//*****
#include<extcode.h>
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<math.h>
struct conj_error{
    float dato_izq;
    float dato_cent;
    float dato_der;
};

```

```

//*****
struct conj_derror{
    float da_izq;
    float da_cent;
    float da_der;
};
//*****

struct conj_salida{
    float d_izq;
    float d_cent;
    float d_der;
};

//*****

struct ganancia{
    float g0;
    float g1;
    float g2;
};

struct ganancia factor;
//*****
//LUGAR PARA DECLARAR PROTOTIPO DE FUNCIONES
//*****

void ganan_error(struct conj_error *direc_uno);
void gain_derror(struct conj_derror *direc_dos);
void gane_sal(struct conj_salida *direc_tres);
void fuzzy_inputs(float x,float y,struct conj_error pun_uno[8],struct conj_derror
pun_dos[8]);
void asig_fuzzy(struct conj_error apun_uno[8],struct conj_derror
apun_dos[8],struct conj_salida apun_tres[8]);

```

```

void fusificar_e(float valor,float canch[8],float lanch[8],float ranch[8]);
void fusificar_d(float valor,float canch[8],float lanch[8],float ranch[8]);
float maximo(float a,float b);
float minimo(float a,float b);
float saturada_izq(float u,float c,float w);
float saturada_der(float u,float c,float w);
float lado_izq(float u,float c,float w);
float lado_der(float u,float c,float w);
void cursor_inicial(float cur_e[8],float cur_d[8]);
float defusificacion(int n_conjuncion,int n_defusificacion,int n_implicacion );
//*****
/**DECLARACION                                DE                                VARIABLES
GLOBALES*****
float eerror,der_error;
float valor_real;
float ecentro[8];
float eancho[8];
float ehwl[8];
float ehwr[8];
float eu[8];
//*****
float dcentro[8];
float dancho[8];
float dhwl[8];
float dhwr[8];
float du[8];
//*****
float sal_centro[8];
float sal_ancho[8];
//*****
int position[3];
int eureka[5];
int fam_matrix[8][8];

```

```

float numero_f=7.0;
int n_conjuncion;
int n_defusificacion;
int n_implicacion;

//*****

//*****
//INICIO DE LA FUNCION PRINCIPAL MAIN
//*****
//*****
//*****
//*****
//***** main()**
BOOL WINAPI DIIMain (HANDLE hDLL, DWORD dwReason, LPVOID
lpReserved)
{
    return TRUE;
}
__declspec(dllexport) float fuzzy_control(int conju,int defu,int impli,int tiempo,float
rata,float rata_cambio,__int8 pivot00,__int8 pivot01,__int8 pivot02,__int8
pivot03,__int8 pivot04,__int8 pivot05,__int8 pivot06,__int8 pivot10,__int8
pivot11,__int8 pivot12,__int8 pivot13,__int8 pivot14,__int8 pivot15,__int8
pivot16,__int8 pivot20,__int8 pivot21,__int8 pivot22,__int8 pivot23,__int8
pivot24,__int8 pivot25,__int8 pivot26,__int8 pivot30,__int8 pivot31,__int8
pivot32,__int8 pivot33,__int8 pivot34,__int8 pivot35,__int8 pivot36,__int8
pivot40,__int8 pivot41,__int8 pivot42,__int8 pivot43,__int8 pivot44,__int8
pivot45,__int8 pivot46,__int8 pivot50,__int8 pivot51,__int8 pivot52,__int8
pivot53,__int8 pivot54,__int8 pivot55,__int8 pivot56,__int8 pivot60,__int8
pivot61,__int8 pivot62,__int8 pivot63,__int8 pivot64,__int8 pivot65,__int8
pivot66,float gano,float ganu,float gand)

```

```
{

//*****
//*****

struct conj_error vec_uno[8]={

        {-100.0,-75.0,-50.0},
        {-75.0,-50.0,-25.0},
        {-50.0,-25.0 ,0.00},
        {-25.0,0.00,25.0},
        {0.00,25.0,50.00},
        {25.0 ,50.00,75.0},
        {50.0,75.0,100.0}

        };

//*****
struct conj_derror vec_dos[8]= {

        {-100.0,-75.0,-50.0},
        {-75.0,-50.0,-25.0},
        {-50.0,-25.0 ,0.00},
        {-25.0,0.00,25.0},
        {0.00,25.0,50.00},
        {25.0 ,50.00,75.0},
        {50.0,75.0,100.0}

        };

//*****
struct conj_salida vec_tres[8]={

        {0.0,12.5,25.0},
        {12.5,25.0,37.5},
        {25.0,37.5,50.0},
        {37.5,50.0,62.5},
```



```
{50.0,62.5,75.0},  
{62.5,75.0,87.5},  
{75.0,87.5,100.0}
```

```
};
```

```
int n=0;  
int i,i_lim;  
int j,j_lim;  
float premisa;  
float numerador,denominador;  
numerador=0;  
denominador=0;  
j_lim=(numero_f-1)-position[0];  
i_lim=(numero_f-1)-position[1];
```

```
fam_matrix[0][0]=pivot00;fam_matrix[0][1]=pivot01;fam_matrix[0][2]=pivot02;fam_  
matrix[0][3]=pivot03;  
fam_matrix[0][4]=pivot04;fam_matrix[0][5]=pivot05;fam_matrix[0][6]=pivot06;fam_  
matrix[1][0]=pivot10;  
fam_matrix[1][1]=pivot11;fam_matrix[1][2]=pivot12;fam_matrix[1][3]=pivot13;fam_  
matrix[1][4]=pivot14;  
fam_matrix[1][5]=pivot15;fam_matrix[1][6]=pivot16;fam_matrix[2][0]=pivot20;fam_  
matrix[2][1]=pivot21;  
fam_matrix[2][2]=pivot22;fam_matrix[2][3]=pivot23;fam_matrix[2][4]=pivot24;fam_  
matrix[2][5]=pivot25;  
fam_matrix[2][6]=pivot26;fam_matrix[3][0]=pivot30;fam_matrix[3][1]=pivot31;fam_  
matrix[3][2]=pivot32;
```

```

fam_matrix[3][3]=pivot33;fam_matrix[3][4]=pivot34;fam_matrix[3][5]=pivot35;fam_
matrix[3][6]=pivot36;
fam_matrix[4][0]=pivot40;fam_matrix[4][1]=pivot41;fam_matrix[4][2]=pivot42;fam_
matrix[4][3]=pivot43;
fam_matrix[4][4]=pivot44;fam_matrix[4][5]=pivot45;fam_matrix[4][6]=pivot46;fam_
matrix[5][0]=pivot50;
fam_matrix[5][1]=pivot51;fam_matrix[5][2]=pivot52;fam_matrix[5][3]=pivot53;fam_
matrix[5][4]=pivot54;
fam_matrix[5][5]=pivot55;fam_matrix[5][6]=pivot56;fam_matrix[6][0]=pivot60;fam_
matrix[6][1]=pivot61;
fam_matrix[6][2]=pivot62;fam_matrix[6][3]=pivot63;fam_matrix[6][4]=pivot64;fam_
matrix[6][5]=pivot65;
fam_matrix[6][6]=pivot66;

```

```

//*****

```

```

//*****

```

```

ganan_error(vec_uno);
gain_derror(vec_dos);
gane_sal(vec_tres);

```

```

error=rata;
der_error=rata_cambio;
    n_conjuncion=conju;
n_defusificacion=defu;
n_implicacion=impli;

```

```

fuzzy_inputs( eerror, der_error,vec_uno,vec_dos);
asig_fuzzy(vec_uno,vec_dos,vec_tres);
fusificar_e(error,ecentro,ehwl,ehwr);

```

```
fusificar_d(der_error,dcentro,dhwl,dhwr);
cursor_inicial(eu,du);
```

```

//*****
//*****

if (i_lim>0 && j_lim>0){
    for(j=0;j<2;j++){
        for(i=0;i<2;i++){
            eureka[n]=fam_matrix[position[0]+j][position[1]+i];
            if(n_conjuncion==2) //CONJUNCION POR PRODUCTO
                premisa=eu[position[0]+j]*du[position[1]+i];
            else
                premisa=minimo(eu[position[0]+j],du[position[1]+i]);

            //*****DEFUSIFICACION*****//
            if(n_defusificacion==2){ //COA
                denominador+=premisa;
                numerador+=premisa*sal_centro[eureka[n]];
            }
            else{
                if(n_implicacion==2){
                    denominador+=premisa*sal_ancho[eureka[n]];

                    numerador+=premisa*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
                }
                else{
                    denominador+=(premisa-
premise*premise/2)*sal_ancho[eureka[n]];

```

```

                                numerador+=(premisa-
premisa*premisa/2)*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
                                }

                                }
                                n+=1;

                                }

                                }

                                }

else if (i_lim>0 && j_lim==0){

    for(i=0;i<2;i++){
    eureka[n]=fam_matrix[position[0]][position[1]+i];
    if(n_conjuncion==2)          //CONJUNCION POR PRODUCTO
        premisa=eu[position[0]]*du[position[1]+i];
    else
        premisa=minimo(eu[position[0]],du[position[1]+i]);
    //*****DEFUSIFICACION*****//
        if(n_defusificacion==2){          //COA
            denominador+=premisa;
            numerador+=premisa*sal_centro[eureka[n]];
        }
        else{
            if(n_implicacion==2){
                denominador+=premisa*sal_ancho[eureka[n]];

numerador+=premisa*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
            }

```

```

                                else{
                                    denominador+=(premisa-
premise*premise/2)*sal_ancho[eureka[n]];
                                    numerador+=(premisa-
premise*premise/2)*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
                                }

                                }
                                n+=1;

                                }

                                }

else if (i_lim==0 && j_lim>0){
    for(j=0;j<2;j++){

        eureka[n]=fam_matrix[position[0]+j][position[1]];
        if(n_conjuncion==2)           //CONJUNCION POR PRODUCTO
            premisa=eu[position[0]+j]*du[position[1]];
        else
            premisa=minimo(eu[position[0]+j],du[position[1]]);
        //*****DEFUSIFICACION*****//
        if(n_defusificacion==2){           //COA
            denominador+=premisa;
            numerador+=premisa*sal_centro[eureka[n]];
        }
        else{
            if(n_implicacion==2){

```

```

        denominador+=premisa*sal_ancho[eureka[n]];

    numerador+=premisa*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
    }
    else{
        denominador+=(premisa-
    premisa*premisa/2)*sal_ancho[eureka[n]];
        numerador+=(premisa-
    premisa*premisa/2)*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
    }

    }
    n+=1;

}

}

else{

    eureka[n]=fam_matrix[position[0]][position[1]];
    if(n_conjuncion==2) //CONJUNCION POR PRODUCTO
        premisa=eu[position[0]]*du[position[1]];
    else
        premisa=minimo(eu[position[0]],du[position[1]]);
    /****DEFUSIFICACION***/
    if(n_defusificacion==2){ //COA

```

```
        denominador+=premisa;
        numerador+=premisa*sal_centro[eureka[n]];
    }
    else{
        if(n_implicacion==2){
            denominador+=premisa*sal_ancho[eureka[n]];

            numerador+=premisa*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
        }
        else{
            denominador+=(premisa-
            premisa*premisa/2)*sal_ancho[eureka[n]];
            numerador+=(premisa-
            premisa*premisa/2)*sal_ancho[eureka[n]]*sal_centro[eureka[n]];
        }
    }

}

return numerador/denominador;
```

```
/*******
```

```
//VISUALIZACION DE LOS VALORES DE LAS FUNCIONES DESPUES DE LA
LLAMADA
```

```
//A FUNCIONES,PASO POR DIRECCION
```

```
//*****
```

```
//*****
```

```
}
```

```
//*****
```

```
//*****
```

```
//*****
```

```
//*****
```

```
//*****
```

```
void ganan_error (struct conj_error *direc_uno)
```

```
{
```

```
int k;
```

```
float aux,aux_uno,aux_dos;
```

```
for(k=0;k<=6;k++)
```

```
{
```

```
    aux=(direc_uno->dato_izq)*factor.g0 ;
```

```
    direc_uno->dato_izq=aux;
```

```
    printf("\n\tconj_valor %f\n",direc_uno->dato_izq);
```

```
    aux_uno=(direc_uno->dato_cent)*factor.g0 ;
```

```
    direc_uno->dato_cent=aux_uno;
```

```
    printf("\n\tconj_valor %f\n",direc_uno->dato_cent);
```

```
    aux_dos=(direc_uno->dato_der)*factor.g0 ;
```

```
    direc_uno->dato_der=aux_dos;
```

```
    printf("\n\tconj_valor %f\n",direc_uno->dato_der);
```

```
    getch();
```

```
    ++direc_uno;
```



```

    }

}

//*****
//*****

void gain_derror(struct conj_derror *direc_dos)
{
int i;
float aux,aux_uno,aux_dos;

    for(i=0;i<=6;i++)
    {
//      clrscr();
      aux=(direc_dos->da_izq)*factor.g1 ;
      direc_dos->da_izq=aux;
      printf("\n\tconj_valor %f\n",direc_dos->da_izq);
      aux_uno=(direc_dos->da_cent)*factor.g1 ;
      direc_dos->da_cent=aux_uno;
      printf("\n\tconj_valor %f\n",direc_dos->da_cent);
      aux_dos=(direc_dos->da_der)*factor.g1 ;
      direc_dos->da_der=aux_dos;
      printf("\n\tconj_valor %f\n",direc_dos->da_der);
      getch();
      ++direc_dos;

    }

}

//*****
//*****

void gane_sal(struct conj_salida *direc_tres)
{

```

```

int m;
float aux,aux_uno,aux_dos;

    for(m=0;m<=6;m++)
    {
//      clrscr();
      aux=(direc_tres->d_izq)*factor.g2 ;
      direc_tres->d_izq=aux;
      printf("\n\tconj_valor %f\n",direc_tres->d_izq);
      aux_uno=(direc_tres->d_cent)*factor.g2 ;
      direc_tres->d_cent=aux_uno;
      printf("\n\tconj_valor %f\n",direc_tres->d_cent);
      aux_dos=(direc_tres->d_der)*factor.g2 ;
      direc_tres->d_der=aux_dos;
      printf("\n\tconj_valor %f\n",direc_tres->d_der);
      getch();
      ++direc_tres;

    }
}

//*****
*
//*****

void fuzzy_inputs(float x,float y,struct conj_error pun_uno[8],struct conj_derror
pun_dos[8])
{
//int i;
    if(x<pun_uno[0].dato_izq){
        eerror=pun_uno[0].dato_izq;
    }
    if(x>pun_uno[6].dato_der){
        eerror=pun_uno[6].dato_der;

```

```

    }
    if(y<pun_dos[0].da_izq){
        der_error=pun_dos[0].da_izq;
    }
    if(y>pun_dos[6].da_der){
        der_error=pun_dos[6].da_der;
    }
    printf ("ERROR ES IGUAL A %f",eerror);
    printf ("DER_ERROR ES IGUAL A %f",der_error);
    getch();

}
//*****
//*****

void    asig_fuzzy(struct    conj_error    apun_uno[8],struct    conj_derror
apun_dos[8],struct conj_salida apun_tres[8])
{
int i;

    for(i=0;i<=6;i++){
        ecentro[i]=apun_uno[i].dato_cent;
    }
    for(i=0;i<=6;i++){
        eancho[i]=apun_uno[i].dato_der-apun_uno[i].dato_izq;
    }
    for(i=0;i<=6;i++){
        ehwl[i]=apun_uno[i].dato_cent-apun_uno[i].dato_izq;
    }
    for(i=0;i<=6;i++){
        ehwr[i]=apun_uno[i].dato_der-apun_uno[i].dato_cent;
    }
//*****

```

```

for(i=0;i<=6;i++){
dcentro[i]=apun_dos[i].da_cent;
}
for(i=0;i<=6;i++){
dancho[i]=apun_dos[i].da_der-apun_dos[i].da_izq;
}
for(i=0;i<=6;i++){
dhwl[i]=apun_dos[i].da_cent-apun_dos[i].da_izq;
}
for(i=0;i<=6;i++){
dhwr[i]=apun_dos[i].da_der-apun_dos[i].da_cent;
}
//*****

for(i=0;i<=6;i++){
sal_centro[i]=apun_tres[i].d_cent;
}
for(i=0;i<=6;i++){
sal_ancho[i]=apun_tres[i].d_der-apun_tres[i].d_izq;
}

}
//*****
//*****

void fusificar_e(float valor,float canch[7],float lanch[7],float ranch[7])

{
int i;
int t=numero_f;
eu[0]=saturada_izq(valor,canch[0],ranch[0]);
eu[t-1]=saturada_der(valor,canch[t-1],lanch[t-1]);
for(i=1;i<t-1;i++){
if(valor<canch[i])

```

```

        eu[i]=lado_izq(valor,canch[i],lanch[i]);
    else
        eu[i]=lado_der(valor,canch[i],ranch[i+1]);

    }

}

//*****
//*****
void fusificar_d(float valor,float canch[7],float lanch[7],float ranch[7])
{
int i;
int t=numero_f;
du[0]=saturada_izq(valor,canch[0],ranch[0]);
du[t-1]=saturada_der(valor,canch[t-1],lanch[t-1]);
    for(i=1;i<t-1;i++){
        if(valor<canch[i])
            du[i]=lado_izq(valor,canch[i],lanch[i]);
        else
            du[i]=lado_der(valor,canch[i],ranch[i+1]);

    }

}

//*****
float maximo(float a,float b)
{
return (a>b?a:b);
}

```

```
//*****  
float minimo(float a,float b)  
{  
return (a<b?a:b);  
}  
//*****  
float saturada_izq(float u,float c,float w)  
{  
    if(u<c)  
        return 1.0;  
    else  
        return maximo(0.0,(1.0-(u-c)/w));  
}  
//*****  
float saturada_der(float u,float c,float w)  
{  
    if(u>c)  
        return 1.0;  
    else  
        return maximo(0.0,(1.0-(c-u)/w));  
}  
//*****  
float lado_izq(float u,float c,float w)  
{  
return maximo(0.0,(1.0-(c-u)/w));  
}  
//*****  
float lado_der(float u,float c,float w)  
{  
return maximo(0.0,(1.0-(u-c)/w));  
}
```

```
}  
//*****  
void cursor_inicial(float cur_e[7],float cur_d[7])  
{  
int i;  
    for(i=0;i<numero_f;i++){  
        if(cur_e[i]!=0.0){  
            position[0]=i;  
            break;  
        }  
    }  
    //*****  
    for(i=0;i<numero_f;i++){  
        if(cur_d[i]!=0.0){  
            position[1]=i;  
            break;  
        }  
    }  
  
}  
//*****
```

B.2 PROGRAMA EN LABVIEW

Por ser el programa muy extenso, se muestran las partes principales del programa:

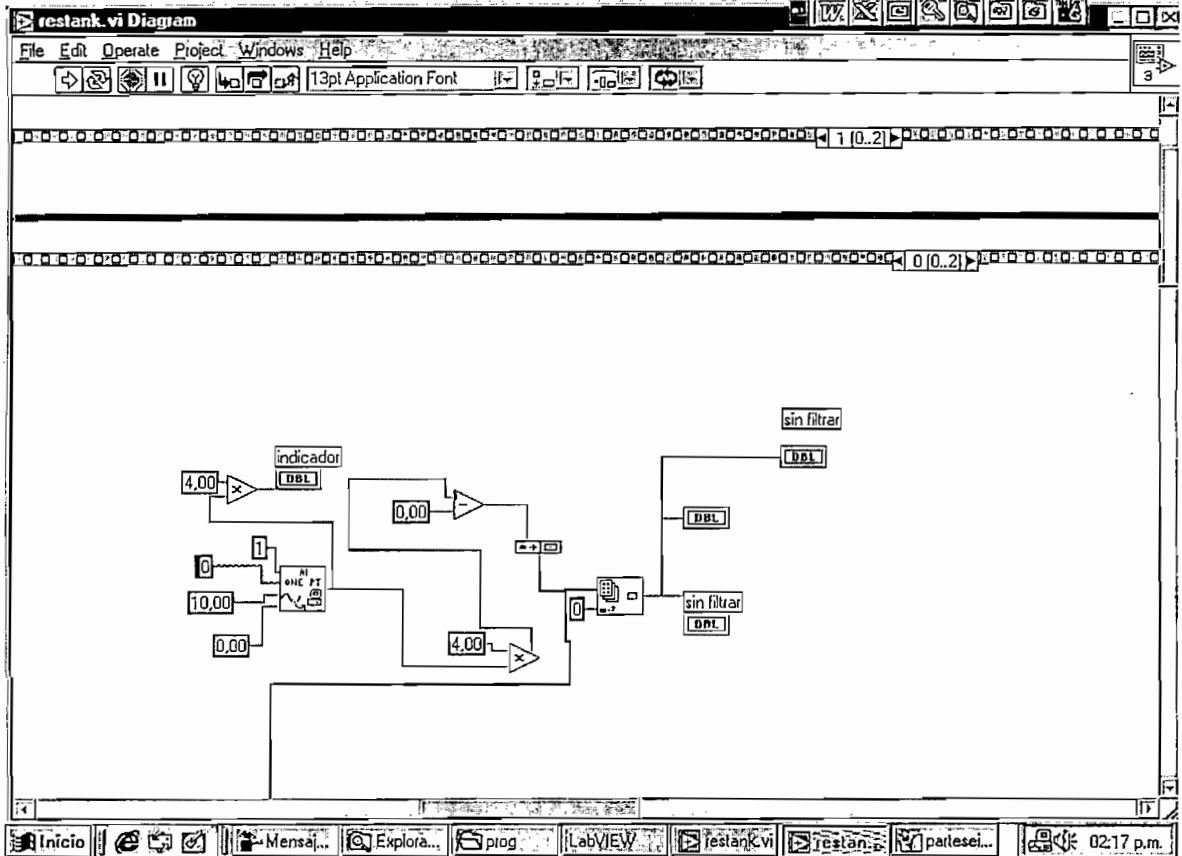


Figura b1. Muestra la adquisición de datos.

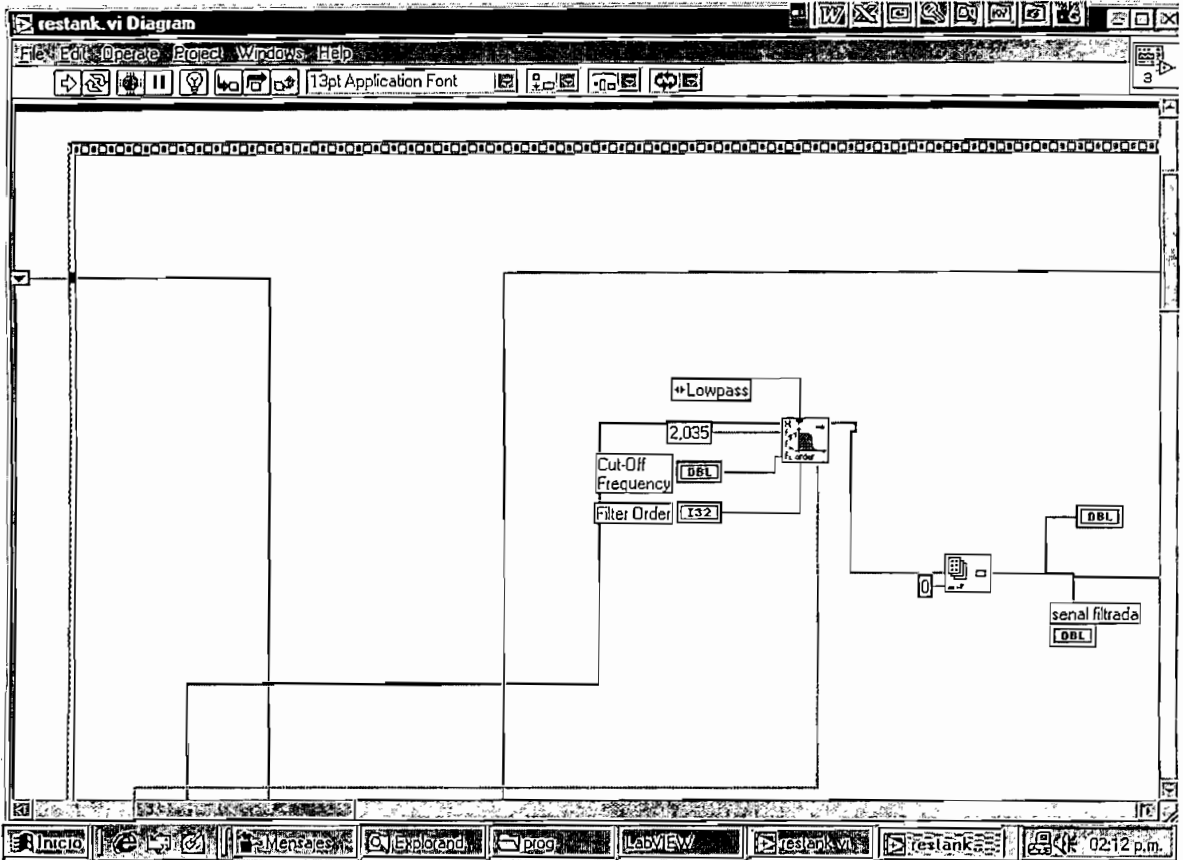


Figura b2.Muestra el filtrado digital de la señal.

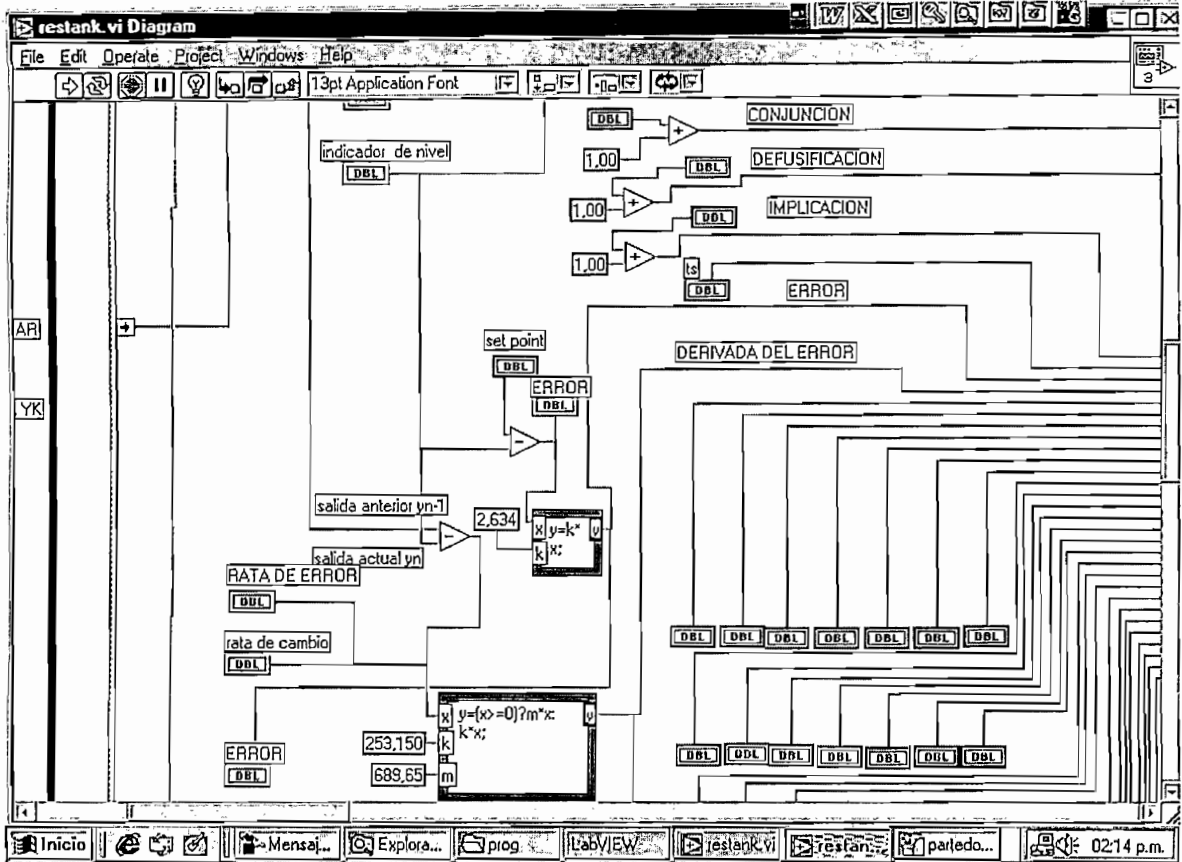


Figura b3. Muestran algunas entradas de datos.

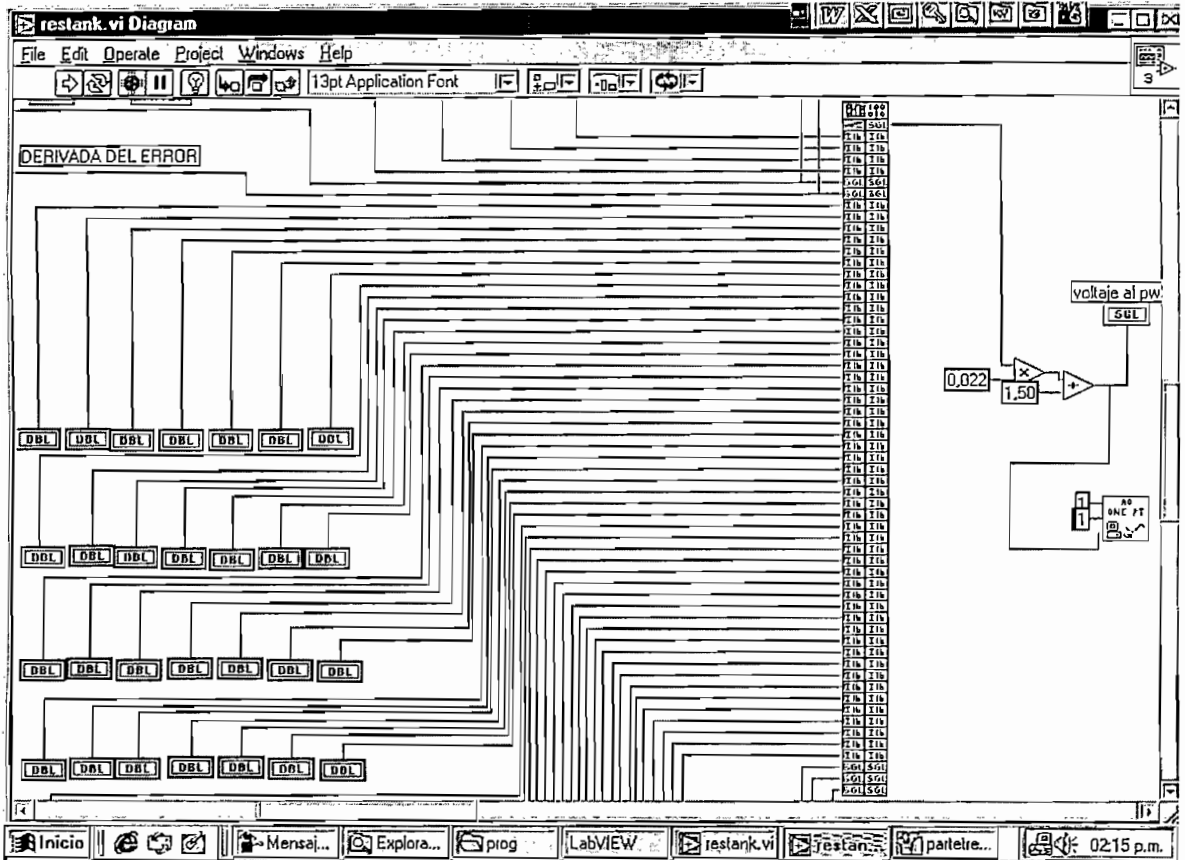


Figura b4. Adquisición de las reglas de inferencia.

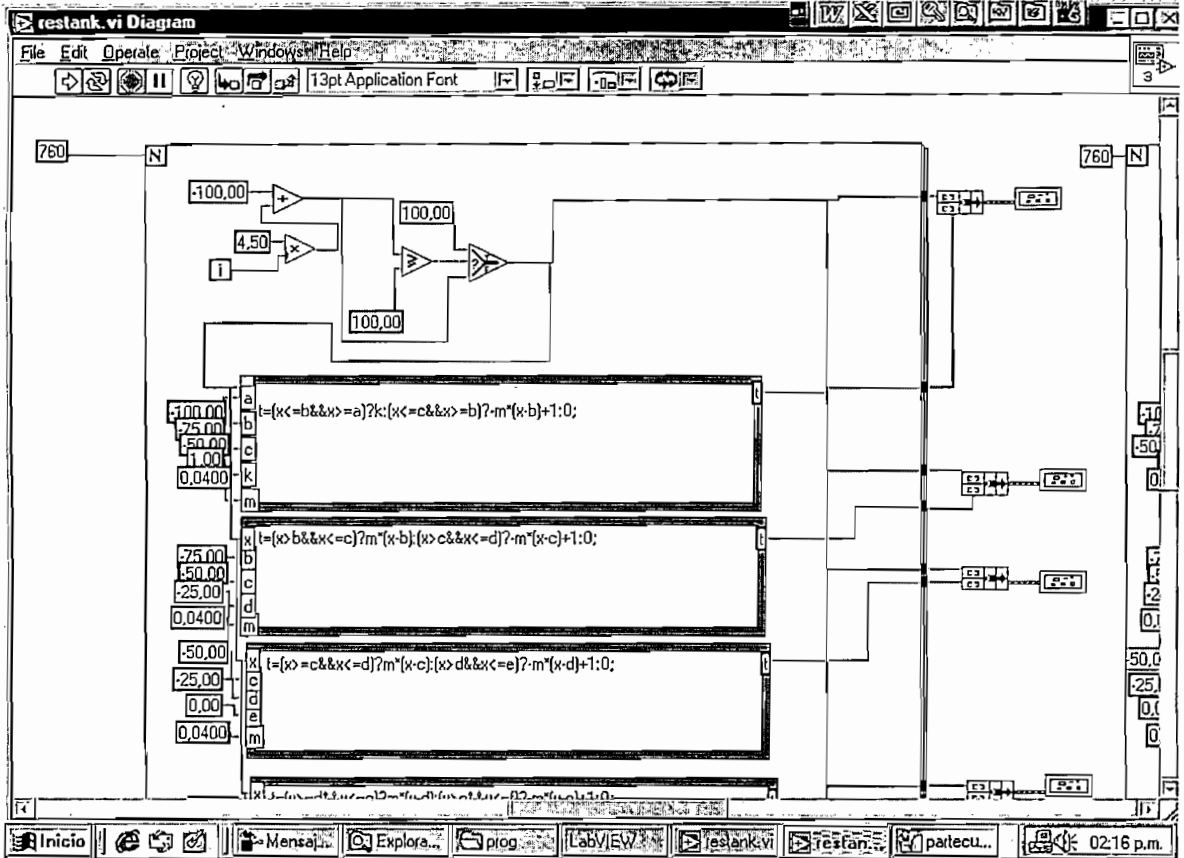


Figura b5. Generación de las funciones de pertenencia.