

**UNIVERSIDAD NACIONAL DEL CALLAO**  
**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**



**TESIS**

**“ALGORITMO GENETICO APLICADO AL CONTROL  
DIFUSO DE POSICION DE UN SERVOMOTOR”**

**PARA OBTENER EL TITULO PROFESIONAL DE INGENIERO  
ELECTRÓNICO**

**PRESENTADO POR LOS BACHILLERES:**

- ✓ **ORTIZ DOMÍNGUEZ, ALEXANDER YEFFERSON**
- ✓ **VIGO HERNANDEZ , HANS DAYNER**
- ✓ **VARGAS-MACHUCA RAMÍREZ, LUIS M.**

**Callao, 2017**

**PERU**

**UNIVERSIDAD NACIONAL DEL CALLAO**  
**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**



**TESIS**

**“ALGORITMO GENÉTICO APLICADO AL CONTROL DIFUSO DE POSICIÓN  
DE UN SERVOMOTOR”**

**PARA OBTENER EL TÍTULO PROFESIONAL DE INGENIERO  
ELECTRÓNICO**

**PRESENTADO POR LOS BACHILLERES:**

- ORTIZ DOMÍNGUEZ, ALEXANDER Y.
- VIGO HERNÁNDEZ, HANS DAYNER.
- VARGAS-MACHUCA RAMÍREZ, LUIS M.

**ASESOR: Dr. Ing. SANTIAGO LINDER RUBIÑOS JIMÉNEZ**  
**CALIFICACION: 15 (QUINCE)**

-----  
**Ing. LUIS E. CRUZADO MONTAÑEZ**  
**PRESIDENTE DE JURADO**

-----  
**Ing. ABILIO B. CUZCANO RIVAS**  
**SECRETARIO**

-----  
**Ing. ARLICH J. PORTILLO ALLENDE**  
**VOCAL**

**Callao, PERU 2017**

## INDICE

|   |              |
|---|--------------|
| INTRODUCCION .....  | 3            |
| <b>I.PLANTEAMIENTO DEL PROBLEMA</b> .....                 | <b>4</b>     |
| 1.1. Determinación del Problema .....                     | 5            |
| 1.2. Formulación del Problema.....                        | 6            |
| 1.3. Objetivos de la Investigación.....                   | 6            |
| 1.4. Justificación de la Investigación .....              | 7            |
| <b>II MARCO TEORICO</b> .....                             | <b>7</b>     |
| 2.1. Antecedentes del Estudio .....                       | 7            |
| 2.2. Marco Teórico .....                                  | 9            |
| 2.3 Definición de Términos Básicos .....                  | 10           |
| <b>III VARIABLES</b> .....                                | <b>18</b>    |
| 3.1. Definición de la Variables .....                     | 18           |
| 3.2. Operacionalización de las Variables.....             | 19           |
| 3.3. Hipótesis.....                                       | 20           |
| <b>IV METODOLOGIA</b> .....                               | <b>25</b>    |
| 4.1. Tipo de Investigación .....                          | 25           |
| 4.2. Población y Muestra .....                            | 27           |
| 4.3. Técnicas e Instrumentos de Recolección de Datos..... | 30           |
| 4.4. Plan de Análisis Estadísticos de Datos .....         | 33           |
| <b>V HIPOTESIS Y RESULTADOS</b> .....                     | <b>34-52</b> |
| <b>VI. CONCLUSIONES</b> .....                             | <b>58</b>    |
| <b>VII. REFERENCIAS BIBLIOGRAFICAS</b> .....              | <b>65</b>    |

**VIII. MATRIZ DE CONSISTENCIA .....67**

## INTRODUCCION

El control difuso incorpora conocimiento experto en su diseño. Sin embargo, se pueden utilizar métodos basados en optimización para el diseño de los mismos; Como es sabido los algoritmos genéticos son una alternativa importante para resolver problemas de optimización [11, 14, 15 y 23]. Los métodos de sintonización de controladores difusos, planteados como problemas de optimización, pueden aplicarse en los siguientes casos: Sintonización de las funciones de membresía o el diseño de la base de reglas. En esta investigación, se usa un algoritmo genético simple para obtener la base de reglas de inferencia difusa. Uno de los objetivos principales es comparar el desempeño de un controlador difuso, sintonizado en base a la experiencia, contra un controlador diseñado con el método de optimización. Una de las ventajas de usar este método para determinar la base de reglas de inferencia, es que sólo se requiere evaluar una función objetivo, y aplicar los operadores genéticos, que permita al algoritmo una búsqueda aleatoria en todo el espacio factible, asegurando la existencia de una solución, que posiblemente sea la solución óptima. En esencia, optimizar las reglas de inferencia de un controlador lógico difuso (CLD), es encontrar la mejor combinación entre las variables difusas de entrada y de salida, para un determinado rango de operación del CLD. Karr fue uno de los primeros investigadores en usar algoritmos genéticos para sintonizar sistemas difusos [12].

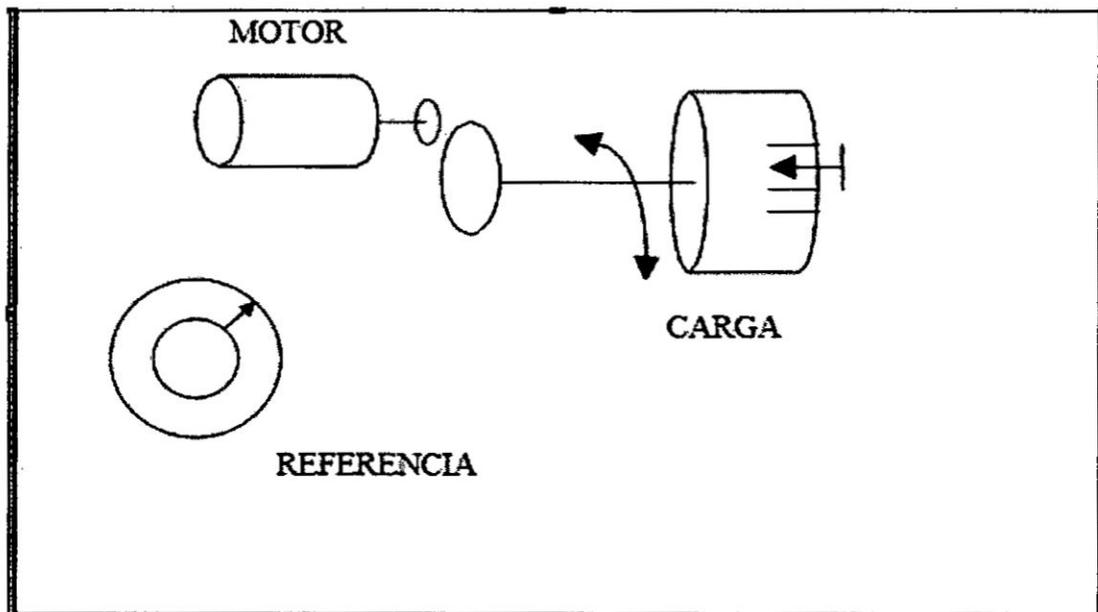
Él propuso un enfoque fuera de línea, para sintonizar las funciones de membresía de un sistema difuso, sin embargo no incluyó la base de reglas en el procedimiento propuesto. Desde su punto de vista la base de reglas obtenidas empíricamente por un humano experto, puede cubrir un rango de operación más amplio. Recientemente, en 2006, Zhang y Li, diseñaron un controlador difuso usando algoritmos genéticos, para controlar la temperatura de un horno. Ellos emplearon un algoritmo genético para obtener la base de reglas del controlador [11].

# CAPITULO I

## 1. PLANTEAMIENTO DEL PROBLEMA

El control de posición considerado en esta tesis, consiste en desplazar de forma angular el eje de un motor de corriente directa, hacia una posición deseada.

La posición se mide en una carga (Inercia), de forma cilíndrica, acoplada al eje del motor por medio de un tren de engranes como se muestra en la figura 1.1.



**Figura 1.1 Servomecanismo**

La carga mueve al eje de un potenciómetro que sirve para proporcionar la variable de posición. El módulo también cuenta con un tacómetro, que nos permite medir la velocidad angular.

Las variables de estado posición y velocidad son utilizadas para el control de posición por el controlador, formándose en ambos casos las señales de error y cambio de error.

El diseño de un controlador lógico difuso para el control de posición de un motor de corriente directa no es novedad actualmente, ya que anteriormente se han realizado trabajos tanto simulados en computadoras como aplicaciones reales de controladores difusos, lo relevante de esta tesis es el diseño y la implantación práctica de un controlador difuso, pero utilizando un algoritmo genético simple para sintonizar las reglas del controlador y poder obtener resultados satisfactorios de desempeño en el dominio del tiempo en cuanto a su tiempo de crecimiento, tiempo de establecimiento y máximo sobrepaso.

Lo significativo de este trabajo es proyectar que la aplicación de los algoritmos genéticos en la lógica difusa, es de gran importancia para sintonizar controladores lógicos difusos de servomotores y logrando así obtener controladores óptimos.

### **1.1. DETERMINACION DEL PROBLEMA**

Las aplicaciones de los controladores basados en Lógica Difusa, han encontrado mucha aceptación por parte de investigadores e Ingenieros, en todos los ámbitos del Control automático. Se ha demostrado en la práctica, que el uso de los CLD's se justifica cuando los procesos a controlar resultan muy complejos o cuando se tienen serias deficiencias en el modelado del mismo. Sin embargo en muchas ocasiones los procesos son tan complejos que la experiencia del diseñador del CLD, por sí sola, no es suficiente para que el CLD diseñado funcione óptimamente. Para mejorar el desempeño de los CLD's, se están empezando a usar otras herramientas, como son: Las Redes Neuronales y los algoritmos genéticos, usados para encontrar una base de conocimiento óptima. En esta investigación se usa un algoritmo genético simple para sintonizar un controlador lógico difuso.

## **1.2. FORMULACION DEL PROBLEMA**

Específicamente se pretende responder a la pregunta: **¿es posible sintonizar un controlador de lógica difusa, aplicando un problema de control de posición de un servomotor?**

## **1.3. OBJETIVOS DE LA INVESTIGACION**

Sintonizar un controlador basado en Lógica difusa, aplicado a un problema de control de posición de un servomotor experimental de corriente directa, Mediante un algoritmo

### **1.3.1 OBJETIVO GENERAL**

Controlar las variables del proceso del sistema algoritmo genético aplicado al control difuso de posición de un servomotor disponible mediante técnicas de control moderno y técnicas de control clásico, para observar el desempeño dinámico de cada uno de los algoritmos implementados.

### **1.3.2 OBJETIVO ESPECÍFICO**

Plantear las etapas del proceso de sistema algoritmo genético aplicado al control difuso de posición de un servomotor con un modelo matemático que la caracterice.

Diseñar algoritmos de control mediante técnicas modernas de control como lógica difusa y optimización por medio de algoritmos genéticos y técnicas clásicas de control como (PID) para la etapa seleccionada del proceso.

Simular y verificar el comportamiento y validez de los controladores diseñados.

Definir las métricas (por ejemplo error estado estacionario) para la comparación del desempeño dinámico de los controladores, y aplicarlas para cada uno de los controladores desarrollados en la estación seleccionada.

## **1.4- JUSTIFICACION DE LA INVESTIGACION**

El algoritmo genético simple será utilizado únicamente para sintonizar las reglas del controlador lógico difuso (CLD). En cuanto a las funciones de membresía de entrada y de salida, serán propuestas en base a conocimiento del proceso a controlar.

# **CAPITULO II**

## **II MARCO TEORICO**

### **2.1. ANTECEDENTES DEL ESTUDIO**

Esta filosofía de control es muy parecida a la forma de pensamiento humano, es decir, está concebido para actuar de una manera similar a como lo haría una persona que conozca determinado proceso y basándose en su experiencia lo pueda controlar. Las ventajas de aplicar estas acciones basadas en el conocimiento y la experiencia por medio del control lógico difuso son: la velocidad con la que se puede realizar así como el número de repeticiones; ya que esta estrategia es implantada en computadoras, microcontroladores o controladores difusos en circuitos integrados.

El control lógico difuso, basado en lógica difusa, tiene como primer antecedente el documento del Dr. Lotfi Zadeh escrito en 1973, en el cual presenta una estrategia nueva para analizar sistemas complejos comparables a los sistemas humanísticos o biológicos [29]. En él hace mención que las técnicas cuantitativas convencionales resultan inapropiadas para analizar ciertos tipos de sistemas, como los referidos anteriormente.

La base de este enfoque es la premisa de que el elemento clave del pensamiento humano no son números, más bien etiquetas de conjuntos difusos, esto es, clases de objetos en los cuales el cambio de pertenencia a no pertenencia a algún conjunto, es gradual y no abrupta como ocurre en los conjuntos tradicionales [27-29].

Mamdani y Assilian [20-21], reportan una aplicación de lógica difusa al control de velocidad de una máquina de vapor, cuyo control con técnicas convencionales era muy deficiente. Esta cita marca el inicio del control lógico difuso presentando una metodología para la realización de controladores difusos, estableciendo claramente los pasos de: fusificación, evaluación de reglas y defusificación.

Después de estas primeras investigaciones ha dado lugar a muchos trabajos relacionados con la lógica difusa, pero es necesario acotar algunas citas, las que más se relacionan con la presente investigación.

En 1989, Y. F. Li y C. C. Lau [18], desarrollaron un algoritmo difuso basado en microprocesador, para controlar posición con un servomotor de corriente directa. Compararon su desempeño con respecto a un controlador PI digital y un control adaptivo por modelo de referencia. Estos resultados se obtuvieron por medio de una simulación hecha en una microcomputadora basada en el microprocesador 6502 y con un lenguaje intérprete de BASIC. Resultando el controlador difuso superior en valores como: error en estado estacionario, tiempo de establecimiento y respuesta transitoria.

En cuanto a controladores difusos sintonizados con métodos de optimización, utilizando algoritmos genéticos o redes neuronales, se tiene como antecedente un trabajo publicado por Karr, en el cual usó algoritmos genéticos para sintonizar las funciones de membresía de un sistema difuso [12]. En el año 2006, Zhang y Li, obtuvieron la base de un controlador lógico difuso mediante algoritmos genéticos, para controlar la temperatura de un horno [11].

Existen herramientas, de dominio público, que sirven para diseñar la base de conocimiento del CLD (Funciones de membresía y reglas), resultando excelentes herramientas cuando el controlador va a operar en un microprocesador o un microcontrolador como es el caso de los microcontroladores MC68hc12.

El MC68hc12 contempla cinco instrucciones específicas relacionadas directamente con los procesos de: fusificación, evaluación de reglas y defusificación, que permiten la aplicación de este microcontrolador en forma natural.

## **2.2. MARCO TEORICO**

### **INTRODUCCIÓN A LA LÓGICA DIFUSA**

La metodología que hace uso de conjuntos difusos definidos por funciones de membresía en expresiones lógicas es llamada Lógica Difusa [26]. Ésta, es una metodología que proporciona una manera simple y elegante de obtener una conclusión a partir de información de entrada vaga, ambigua, imprecisa, con ruido o incompleta, en general la lógica difusa imita como una persona toma decisiones basada en información con las características mencionadas. Una de las ventajas de la lógica difusa es la posibilidad de implementar sistemas basados en ella tanto en hardware como en software o en combinación de ambos.

El concepto de Lógica Difusa fue concebido por Lotfi Zadeh, profesor de la Universidad de California en Berkley, quien inconforme con los conjuntos clásicos (*crisp sets*) que solo permiten dos opciones, la pertenencia o no de un elemento a dicho conjunto, la presentó como una forma de procesar información permitiendo pertenencias parciales a unos conjuntos que en contraposición a los clásicos los denominó Conjuntos Difusos (*fuzzy sets*), el concepto de conjunto difuso fue expuesto por Lotfi Zadeh en un artículo, hoy clásico en la literatura de la lógica difusa, en el año de 1965 [27]. El mismo Zadeh publica en 1971 el artículo, "Quantitative Fuzzy Semantics", en donde Introduce los elementos formales que acabarían componiendo el cuerpo de la doctrina de la lógica difusa y sus aplicaciones tal como se conocen en la actualidad.

El profesor Zadeh menciona que la gente no requiere información numérica precisa del medio que lo rodea para desarrollar tareas de control altamente adaptable por ejemplo conducir un automóvil o caminar por una acera sin chocarse con los

postes y las otras personas. Si los controladores convencionales, en esencia realimentados, se pudieran programar para aceptar entradas con ruido e imprecisas ellos podrían trabajar de una manera más eficiente y quizás se podrían implementar más fácilmente. En Estados Unidos principalmente por razones culturales, el concepto de lógica difusa no tuvo mucho impacto mientras en oriente específicamente los japoneses y algunos países europeos aceptaron sin complicación esta idea y han estado desde la década de los 80's construyendo aplicaciones reales y productos que funcionan basados en lógica difusa. Por ejemplo en 1986 Yamakawa publica el artículo, "Fuzzy Controller Hardware System" y desarrolla controladores difusos en circuitos integrados. En 1987, se inaugura en Japón el subterráneo de Sendai, uno de los más espectaculares sistemas de control difuso creados por el hombre. Desde entonces el controlador inteligente ha mantenido los trenes rodando eficientemente. En 1987, "FUZZY BOOM", se comercializan multitud de productos basados en la lógica difusa (sobre todo en Japón).

## 2.3. DEFINICION DE TERMINOS BASICOS

### 2.3.1. CONJUNTOS DIFUSOS

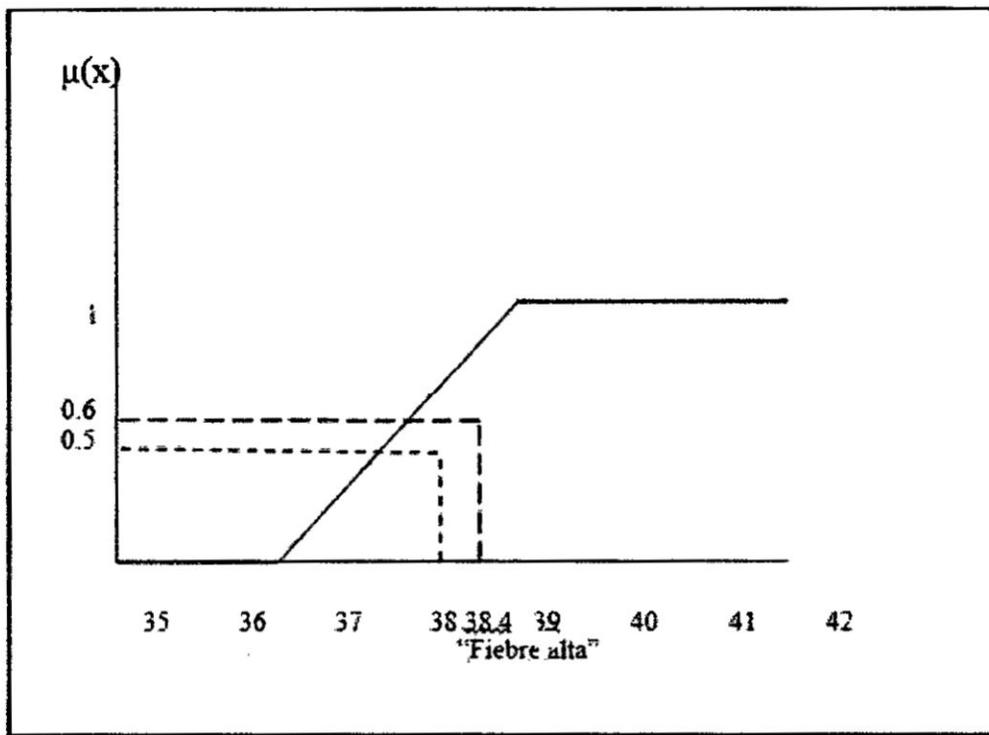
Un conjunto difuso  $A$  de un universo de discurso  $U$  es caracterizado por una función de membresía  $\mu_A: U \rightarrow (0, 1)$ , que asocia a cada elemento  $u$  de  $U$ , un número  $\mu_A(u)$  que puede tomar todos los valores reales comprendidos dentro del intervalo  $(0,1)$  [1], que representa el grado de membresía de  $u$  en  $A$  [20, 21 y 29]. El conjunto difuso  $A$  de  $U = \{u_1, u_2, \dots, u_n\}$  es expresado por:

$$A = \sum_{i=1}^n \mu_A(u_i)/u_i = \sum_i u_A(u_i)$$

Donde  $\sum$  representa *unión*.

En la figura 2.3.1 se presenta un conjunto difuso denominado "Fiebre alta",

donde se puede apreciar el cambio continuo del valor de pertenencia, para cada uno de los puntos del universo de discurso. En esta figura se hace énfasis en dos valores de temperatura cuya diferencia es pequeña, así, las temperaturas 38° y 38.4° se evalúan ligeramente diferente y no como un cambio abrupto, lo cual pudiera ocurrir en un conjunto tradicional.



**Figura 2.3.1 conjunto Difuso, "fiebre alta"**

A continuación se definen los tres operadores básicos utilizados en esta tesis:

$$A + B = \sum_i \mu_A(u_i) \vee \mu_B(u_i)$$

Donde  $\vee$  representa el máximo (Max). La unión corresponde al conectivo OR. (b) La intersección de A y B se representa  $A.B$  y se define por:

$$A.B = \sum_i \mu_A(u_i) \wedge \mu_B(u_i)$$

Donde  $\wedge$  representa el mínimo (min). La intersección corresponde al conectivo AND.

(c) El complemento de un conjunto A se representa  $\Gamma A$  y se define por:

$$\Gamma A = \sum_i 1 - \mu_A(u_i)$$

El complemento corresponde a la negación NOT.

### Funciones de membresía difusa

El grado de membresía en los conjuntos difusos, puede ser representado por una función continua, la cual es denominada como función de membresía. En otras palabras, si  $F$  es un conjunto difuso, entonces la función de membresía  $\mu_F(x)$  mide el grado con el cual el valor  $x$  pertenece al conjunto  $F$ , de la cual puede tomar diferentes formas, dependiendo de la aplicación en particular [10 y 30]. Algunas de las funciones más conocidas son: función *singleton*, cuyo valor de membresía es igual a 1 para un solo punto y el resto es cero, función trapezoidal, triangular, *sigmoidal*, entre otras. La figura 2.3.2 muestra una función de membresía de forma triangular, donde se aprecia que los valores de membresía cambian en forma lineal.

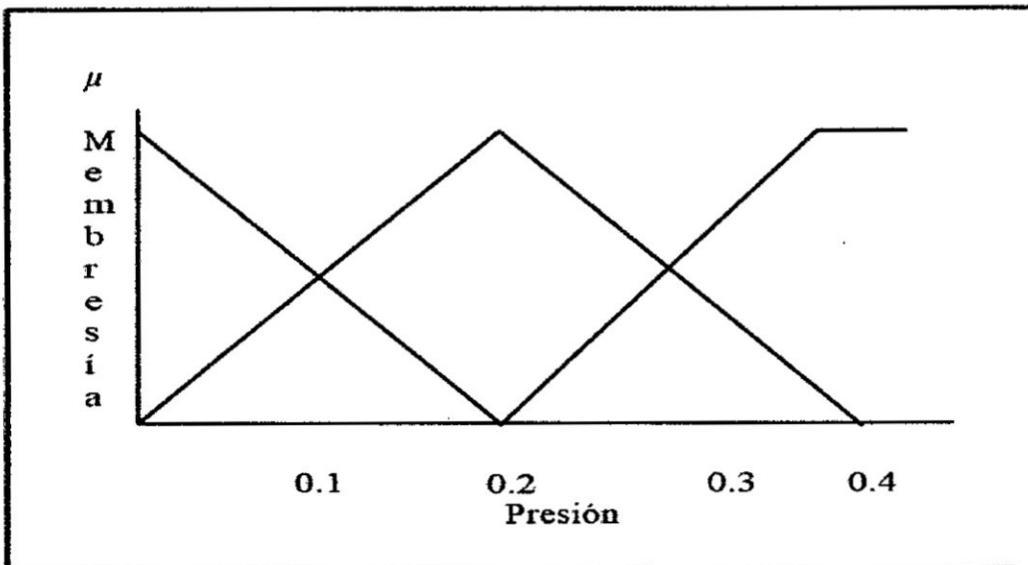
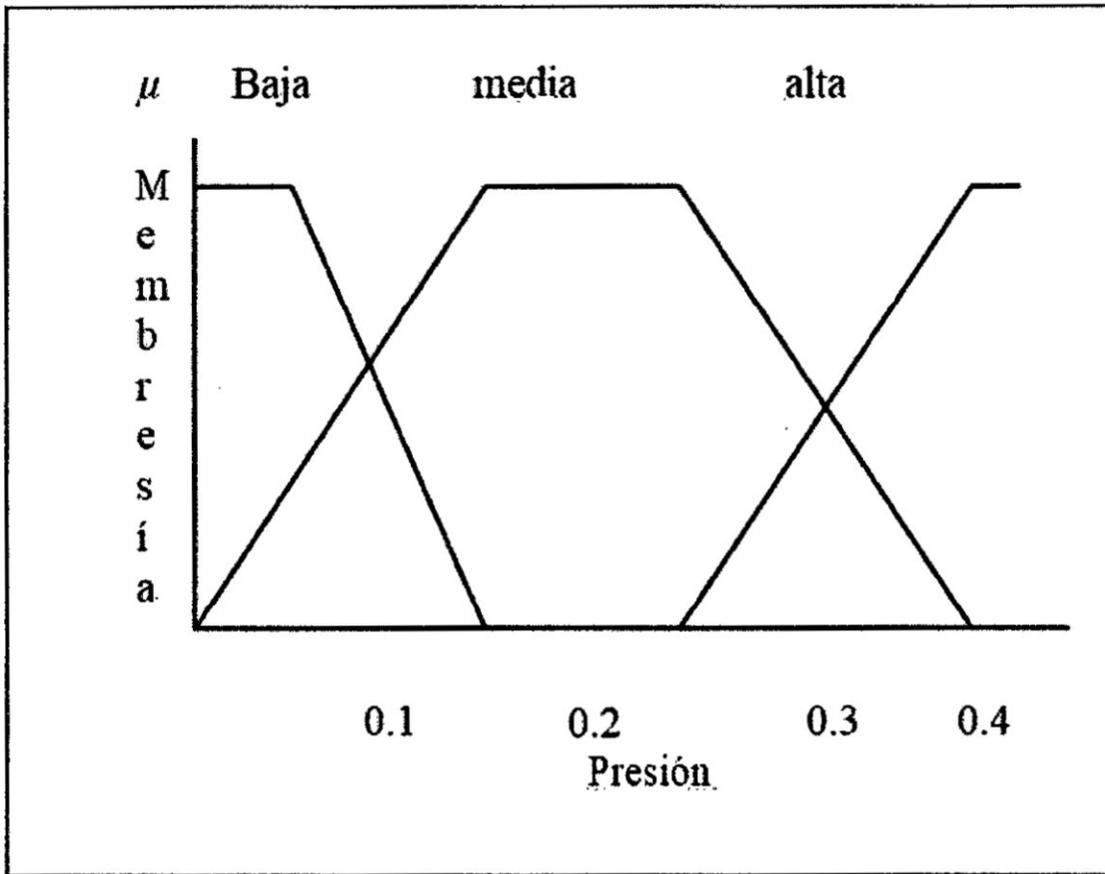
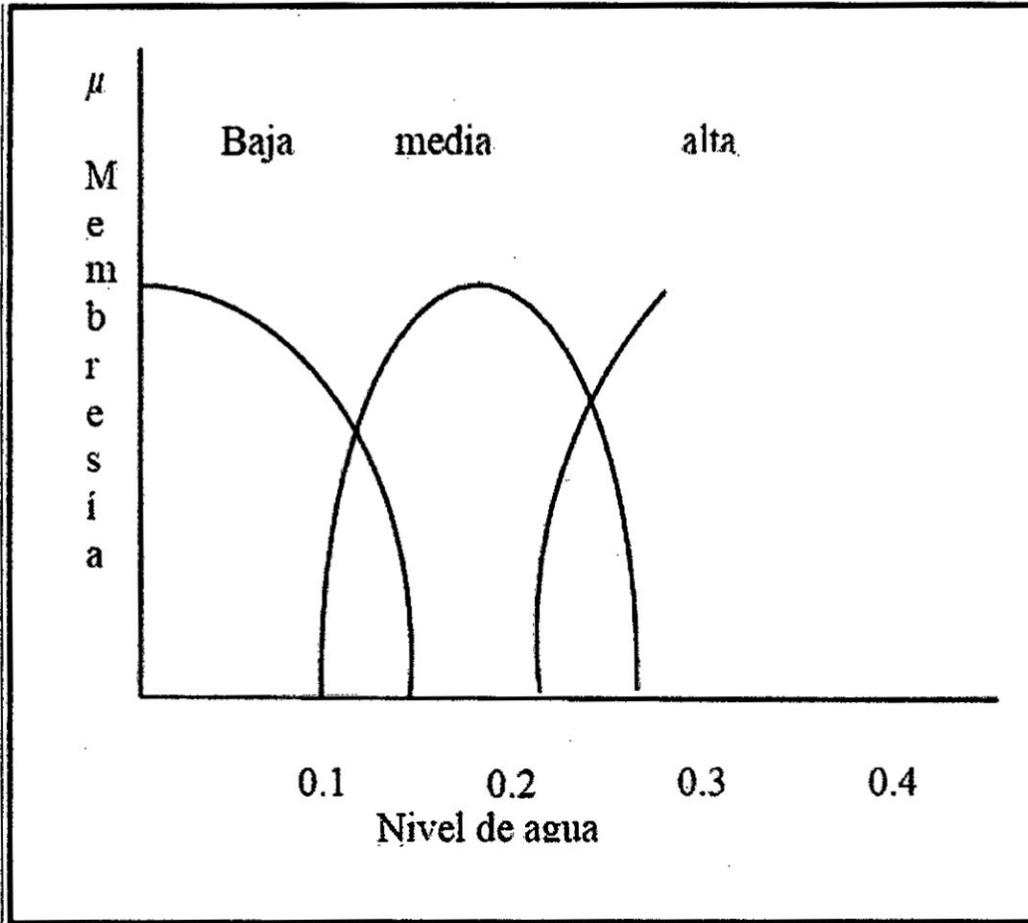


Figura 2.3.2. Función de membresía triangular

En las figuras 2.3.3 y 2.3.4 se presentan otras formas de funciones de membresía, la figura 2.3.3 muestra una función de membresía trapezoidal, y la figura 2.3.4 presenta una función de membresía no-lineal.



**Figura 2.3 .3. Función de membresía trapezoidal**



**Figura 2.3.4. Función de membresía no-lineal**

Una notación convencional para los conjuntos difusos que es popular en la literatura cuando el universo de discurso  $U$ , es discreto y finito, es dado para el conjunto difuso  $A$  mediante la ecuación 6.5:

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots = \sum_I \frac{\mu_A(x_i)}{x_i}$$

y cuando el universo de discurso  $U$  es continuo e infinito, el conjunto difuso  $A$  se representa por la ecuación 6.6:

$$A = \int \frac{\mu_A(x)}{x}$$

### 2.3.2. CONTROLADOR LÓGICO DIFUSO

Un controlador lógico difuso (CLD), emplea principios de lógica difusa y nos permite convertir estrategias de control lingüístico, basado e conocimiento experto, en una estrategia de control automático [2, 24 y 25]. Hasta la fecha no existe una metodología única, sin embargo para efecto de esta tesis se utiliza el método propuesto por C.C. Lee [16-17], el cual está compuesto por las siguientes partes:

- a) Fusificación.
- b) Base de conocimiento.
- c) Lógica de decisiones.
- d) Defusificación.

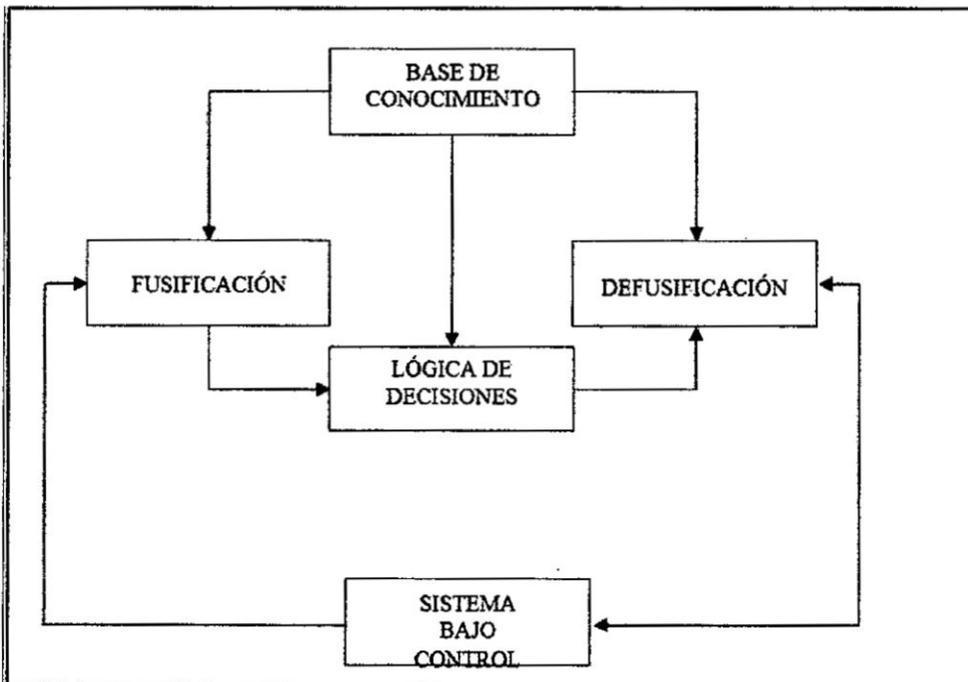


Figura 2.3.2.1 Estructura básica de un controlador lógico difuso

### 2.3.3. FUSIFICACIÓN

Es el proceso de asignar valores de membresía o pertenencia a un valor numérico de entrada para cada una de las etiquetas difusas que forman la variable lingüística; por ejemplo, la variable lingüística "Temperatura de la sala" puede tomar los valores "baja", "semi-baja", "media" y "alta"; para este caso la entrada al fusificador es un valor de temperatura preciso (*crisp*), y la salida estará formada por los valores de verdad de cada una de las etiquetas "baja", "semi-baja", "media" y "alta", como se muestra 2.3.2. en La figura 2.3.3.

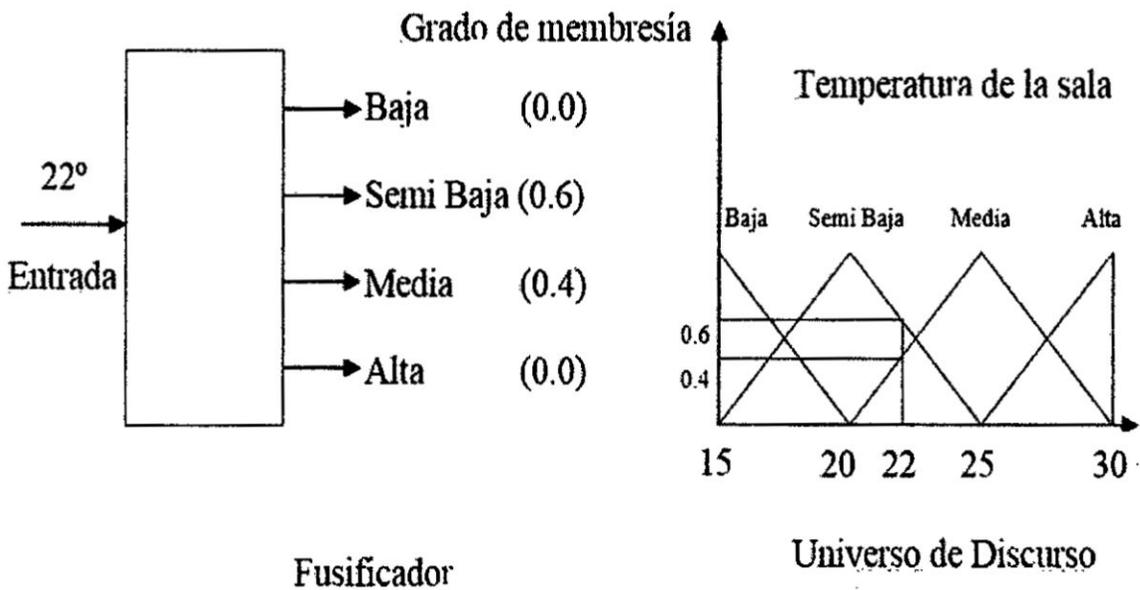


Figura 2.3.3. Proceso de fusificación

#### **2.3.4 BASE DE CONOCIMIENTO**

Se obtiene de la experiencia desarrollada por un operador y de conocimientos sobre Ingeniería de control, y depende del proceso a controlar y los requerimientos de diseño. Está compuesta de dos partes, una base de datos y una base de reglas de control difuso (utiliza variables lingüística). Las bases de reglas también se puede obtener utilizando métodos de optimización [11 y 15].

##### **Base de datos**

Incluye la discretización y normalización de los universos de entrada y salida, la definición de subconjuntos (partición de los universos y funciones de pertenencia) y satisfacer la propiedad de completitud.

##### **a). Discretización.**

Debido a la necesidad de discretizar, ya que los datos se procesan en forma digital, este proceso genera niveles cuantizados, cada uno de los cuales representa un elemento genérico en un universo de discurso.

##### **b). Normalización.**

La normalización del universo discreto puede ser lineal o no. c). Partición de los universos.

Se refiere al número de etiquetas que toma una variable lingüística; por ejemplo la variable "temperatura de la sala", puede tomar los valores "muy baja", "baja", "media" y "alta". El número de términos está determinado por las características del sistema a controlar y la calidad de control.

##### **d). Funciones de pertenencia.**

Como ya se ha mencionado anteriormente las funciones de pertenencia o membresía, representan gráficamente la relación que mantienen los elementos de un subconjunto difuso, dentro de un universo de discurso, con el grado de pertenencia al conjunto en cuestión. La representación de estas figuras puede tomar diferentes formas, siendo las más utilizadas las de forma triangular y trapezoidal. La utilización de

alguna de ellas puede ser de manera arbitraria, dependiendo de la aplicación en particular [30].

**e). Completitud.**

Esta propiedad indica a que el algoritmo debe ser capaz de inferir una acción correcta para cada estado del proceso.

## **CAPITULO III**

### **3.1 DEFINICION DE VARIABLES**

#### **3.1 Variables de la Investigación**

Relacionando las variables relevantes que intervienen en el presente problema objeto de estudio, que conllevan a la explicación, demostración y comprobación de la formulada hipótesis, se han identificado las siguientes variables.

##### **3.1.1 Variable independiente**

La variable independiente, también conocida como variable manipulada, se encuentra en el centro de cualquier diseño experimental cuantitativo. Para nuestro presente trabajo de investigación la variable independiente es: **EI ALGORITMO GENETICO.**

##### **3.1.2 Variable dependiente**

En cualquier experimento real, un investigador manipula una variable independiente para influir en una o varias variables dependientes. Para nuestro presente trabajo de investigación tenemos las siguientes variables dependientes que son el uso de muestras de recolección de datos.

### 3.2 OPERACIONALIZACION DE LAS VARIABLES

La estrategia de control, derivada de la experiencia se expresa mediante el uso de algoritmos difusos. Las reglas de control que forman el algoritmo difuso pueden definirse usando los siguientes criterios:

#### a) Selección de las variables.

Las variables de entrada se seleccionan basándose en la experiencia y en conocimientos de Ingeniería y el cambio de error (derivada del error). La importancia de usar la variación de error se ilustra con un caso cotidiano, como es el hecho de atravesar una avenida, para lo cual no solo consideramos la distancia entre un auto y la persona que va cruzar la calle, sino también consideramos muy relevante la rapidez con la que el auto se desplaza.

#### b) Origen y obtención de las reglas de control.

- La experiencia y los conocimientos en Ingeniería de control. Es la que más se utiliza [18].
- Utilizando métodos de optimización [11 y 15].

En un controlador lógico difuso, su operación dinámica está caracterizada por un conjunto de reglas, compuestas por variables lingüísticas, basada en conocimiento experto es usualmente de la forma:

*IF* (un conjunto de condiciones son satisfechas)

*Then* (un conjunto de consecuentes que pueden inferir).

Donde los antecedentes y los consecuentes de las reglas *IF-Then* son asociados con conceptos difusos (términos lingüísticos), formando lo que se conoce como declaraciones condicional difusa en donde el antecedente es una condición sobre la base del estado de las variables del proceso y el consecuente es una acción de control para el sistema a controlar (proceso).

**c) Tipos de reglas de control.**

Lee C. C. [16], menciona que existen dos tipos de reglas usadas en el diseño de un CLD: las reglas de control de evaluación de estado y las reglas de evaluación de objeto.

**Reglas de evaluación de estado**

Este tipo de reglas es el que más se usa, y en el caso de los sistemas MISO, del inglés entradas múltiples salida única, toman la siguiente forma:

R1: si x es A1,....., y w es B1 entonces z es C1

R2: si x es A2,....., y w es B2 entonces z es C2

R3: si x es A3,....., y w es B3 entonces z es C3.

Rn: si x es An,....., y w es Bn entonces z es Cn.

**Reglas de evaluación de objeto**

Este tipo de reglas involucra la evaluación del estado actual así como el resultado de la acción de control. Se aplica en esquemas de control difuso de tipo predictivo.

**3.3 HIPOTESIS**

**LÓGICA DE DECISIONES**

Como ya se ha mencionado un CLD puede emular a un hábil experto operador humano [16], mediante funciones de implicación difusa, y mecanismos de inferencia principalmente. Aunque no son los únicos mecanismos que existen.

**Inferencia difusa**

Después que las variables de entrada han sido convertidas a valores de variables lingüísticas, el paso de inferencia difusa identifica las reglas que se aplican a cada situación, y mediante un método llamado MAX/MIN [7], determina los valores de la variable lingüística de salida, en el caso que se tenga solo una salida. El

siguiente ejemplo muestra cómo se aplica este método el cual está compuesto de dos pasos:

- Agregación, que relaciona los antecedentes de las reglas.
- Composición, el cual procesa los consecuentes de las mismas.

Suponiendo que se está controlando un proceso y las variables lingüísticas de entrada son: "distancia" y "ángulo" y la variable de salida: "potencia"; se pueden escribir a manera de ejemplo, la siguiente base de reglas con los valores de pertenencia de las etiquetas que componen los antecedentes de cada una de las reglas.

|         |   |
|---------|---|
| Regla 1 | Si distancia = media $(0.9)^{(1)}$ y ángulo = positivo pequeño $(0.8)^{(1)}$<br><b>entonces</b> potencia = positiva media |
| Regla 2 | Si distancia = media $(0.9)^{(1)}$ y ángulo = cero $(0.2)^{(1)}$ <b>entonces</b><br>potencia = positiva cero              |
| Regla 3 | Si distancia = grande $(0.1)^{(1)}$ y ángulo = cero $(0.2)^{(1)}$ <b>entonces</b><br>potencia = positiva media            |

Nota: (1) Son valores arbitrarios usados únicamente para ejemplificar la aplicación de los operadores difusos.

La parte de la regla 1 combina las condiciones "distancia = media" y "ángulo = positivo-pequeño", definiendo la validez de la regla en la situación presente. En lógica convencional, la combinación de las dos condiciones se puede evaluar mediante la función Boleana *AND*. En el caso de la lógica difusa no se puede utilizar la función Boleana *AND*, porque esta no maneja valores intermedios entre 0 y 1. En este caso se definen conectivos lógicos *AND*, *OR* y *NOT* representados en La Tabla 6.1.

|     |   |
|-----|---|
| AND | $\mu_A \cap B(u) = \min\{\mu_A, \mu_B(u)\}$ |
| OR  | $\mu_A \cup B(u) = \max\{\mu_A, \mu_B(u)\}$ |

|     |  |
|-----|--|
| NOT | $\mu_{\bar{A}} \cap B(u) = 1 - \mu_{\bar{A}}(u)$ |
|-----|--|

Tabla 6.1. Operadores Lógicos Difusos

Continuando con el ejemplo, se aplica el Operador *AND* a los antecedentes de las reglas 1, 2 y 3 obteniéndose el siguiente resultado:

|     |                      |
|-----|----------------------|
| AND | Min (0.9; 0.8) = 0.8 |
| OR  | Min (0.9; 0.2) = 0.2 |
| NOT | Min (0.1; 0.2) = 0.1 |

Se aprecia, que las reglas 1 y 3 tienen la misma etiqueta de salida, pero diferentes grados de verdad. En lógica difusa, las reglas que tienen el mismo consecuente se evalúan mediante un operador *OR*, representado mediante el operador matemático "Max". Lo anterior es parte del método *MIN/MAX*, llamado composición, que define los valores difusos de salida, para que posteriormente sean utilizados en el proceso de defusificación. El resultado final del ejemplo es el siguiente:

Salida

Positiva media = Max {0.1; 0.8} = 0.8

Cero = 0.2

#### DEFUSIFICACIÓN

Básicamente este proceso [16-17], es un mapeo de un espacio de acciones de control difuso definido sobre un Universo de discurso de salida, en un espacio de acciones de control no-difuso (valores precisos), siendo muy importante esta Interfaz, ya que la mayoría de las aplicaciones prácticas de control requieren variables numéricas. De las estrategias más usadas de pueden mencionar las siguientes:

- Método del criterio máximo.
- Método del promedio máximo.
- Método del centro de áreas.

### Método del criterio máximo

Da como resultado el punto en el cual la distribución de posibilidades de la acción de control alcanza el valor máximo.

### Método del promedio del máximo

Esta estrategia genera una acción de control la cual es el promedio de todas las acciones de control cuyos grados de pertenencia alcanzan el valor máximo. Concretamente en el caso de un universo discreto, la salida de control puede definirse por medio de la siguiente ecuación:

$$Z_o = \sum_{j=1}^I \frac{w_j}{I}$$

Donde  $w_j$  es el valor central de la etiqueta lingüística "j" para la cual la función de membresía alcanza el máximo valor  $\mu_z(w_j)$  y "I" es el número de etiquetas que alcanzan el valor máximo.

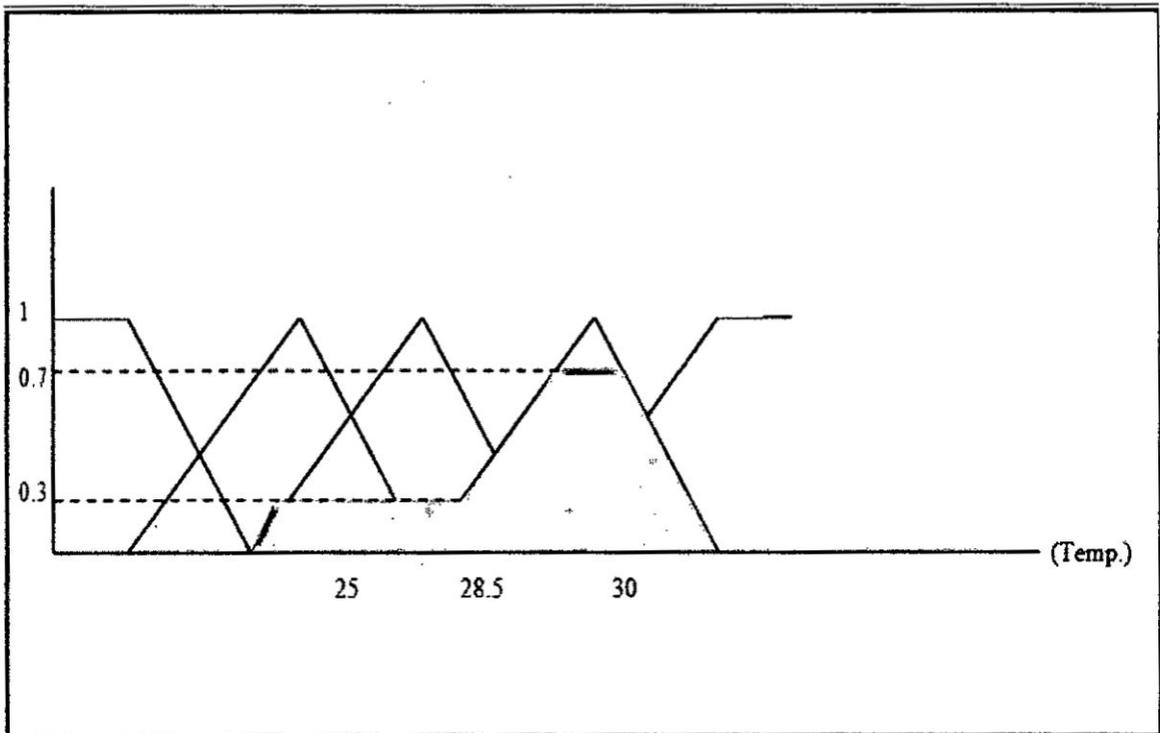
### Método del centro de áreas

Es una de las estrategias más utilizadas [18], se calcula el centro de gravedad de la distribución de las salidas de control difuso para obtener la señal de control, la cual se aplica al sistema a controlar. Para el caso de un Universo discreto, la salida defusificada se puede obtener mediante la siguiente expresión:

$$Z_o = \frac{\sum_{j=1}^n \mu_{zz}(w_j) \cdot w_j}{\sum_{j=1}^n \mu_z(w_j)}$$

Donde  $n$  es el número de niveles de cuantización del universo de salida y  $\mu_z$  es una función de pertenencia de salida resultante. Este método corta la función membresía de

los respectivos términos lingüísticos, de acuerdo con sus valores de pertenencia, formando áreas nuevas las cuales son sobrepuestas para formar una sola. Balanceando esta área se obtiene el valor preciso de salida. Lo anterior se ilustra por medio de la siguiente figura.



**Figura. Método del centro de área**

Como ejemplo de la aplicación de esta estrategia se consideran los datos que se muestran en la figura anterior, y por medio de la siguiente ecuación, se obtiene la salida.

$$Salida = \frac{(0.3)(25) + (0.7)(30)}{0.3 + 0.7}$$

## CAPITULO IV

### 4.1 TIPO DE INVESTIGACION

#### INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS

Los algoritmos genéticos están inspirados en los procesos evolutivos que se presentan en la naturaleza, donde la idea central es la supervivencia de los individuos más aptos y la modificación constante de sus descendientes para adaptarse al entorno donde habitan [5, 6, 9 y 13]. Estos principios son en parte emulados por los algoritmos genéticos, cuando se usan para resolver problemas de optimización [22 y 23]. La técnica de solución de problemas basada en los Algoritmos Genéticos supone que la solución potencial de cualquier problema es un individuo, que puede ser representado por medio de un conjunto de parámetros. Estos parámetros son vistos como los genes de un cromosoma y pueden ser estructurados por medio de una cadena de valores en forma binaria. En control automático, muchos problemas pueden ser planteados como problemas de optimización, con una o más funciones objetivo, sujeto a restricciones. Así por ejemplo, en un problema de control retroalimentado, se podría plantear como función objetivo a la integral del valor absoluto de la señal del error (Índice de desempeño). En este caso, la función objetivo contiene a los parámetros de la planta y a los parámetros del controlador, que son los que se desean encontrar. Cada uno de los individuos de la población, (cromosomas), sobre los cuales opera el algoritmo genético, contiene a los parámetros del controlador que se desea diseñar. Si se tratara de un controlador P-I-D (Proporcional-Integral-Derivativo), entonces cada individuo contendría a las ganancias: proporcional, integral y derivativa. Estos individuos son evaluados por medio de la función objetivo con el fin de asignarles un valor que les califique su aptitud dentro del problema de optimización [14]. Después de que los individuos son evaluados, el proceso continúa con la etapa de selección, donde se mantienen y reproducen los individuos más aptos, para que posteriormente se realicen los procesos genéticos de cruzamiento y mutación, en los cuales los individuos al cruzarse generan descendientes con nuevas características que dan variedad a la población. La mutación ocurre con una tasa de probabilidad baja, por lo que solo algunos de los individuos de la población son alterados. Sin embargo estos pequeños cambios son suficientes para que el

algoritmo pueda explorar todo el espacio de búsqueda y evitar converger hacia un mínimo o un máximo local.

El proceso completo se realiza  $n$  veces con una población inicial, creada en forma aleatoria, con  $m$  individuos. El número de iteraciones, así como el tamaño de la población son propuestas por el usuario. A continuación se detallan cada una de las partes que componen el proceso genético.

### CODIFICACIÓN

En contraste con lo que sucede en estrategia evolutiva y programación evolutiva. Los algoritmos genéticos trabajan con cadenas de bits de extensión fija  $l$ . frecuentemente los algoritmos genéticos se aplican en problemas de control óptimo con parámetros que varían dentro de un rango continuo de valores [5 y 6]. El proceso de codificación consiste en determinar el número de bits para representar a cada parámetro tomando como base los valores mínimo y máximo de los parámetros, así, como la resolución deseada. El número de bits para codificar un parámetro se puede encontrar a partir de la siguiente ecuación:

$$R = \frac{U - V}{2^l - 1}$$

$V$ : mínimo valor del rango del parámetro

$U$ : máximo valor del rango del parámetro

$l$ : número de bits necesarios para codificar el parámetro

$R$ : resolución deseada.

**Ejemplo 4.1. Determinar el número de bits necesarios para codificar un parámetro contenido dentro del intervalo [-5,5] con una resolución de 0.1.**

Aplicando la ecuación 6.9 se tiene como resultado  $l = 6.6582$ , sin embargo, el número de bits debe ser un valor entero, entonces se toma el número inmediato superior, que es en este caso siete, y dado que se modifica el valor de la resolución deseada: el siguiente paso es recalcular el valor de la resolución con el nuevo valor de  $l$ , después del cambio  $l$ , los resultados son los siguientes:

|                   |               |
|-------------------|---------------|
| <i>Resolución</i> | <i>0.0787</i> |
| <i>L</i>          | <i>7 bits</i> |

El resultado anterior se puede visualizar en la Tabla 4.2.

| <b>V. Parámetro Real</b> | <b>V. Codificado</b> |
|--------------------------|----------------------|
| -5                       | 0000000              |
| -4.9213                  | 0000001              |
| -4.8426                  | 0000010              |
| ⋮                        | ⋮                    |
| 5                        | 1111111              |

**Tabla 6.2. Resultado del Ejemplo 4.1**

El número total de bits que conforman cada una de las cadenas binarias es la suma de los números de bits de cada parámetro, conocido como ancho de la cadena. El número de bits por cada parámetro y el ancho de la cadena serán necesarios para inicializar la población, que es el siguiente paso del proceso genético.

## **4.2 POBLACION Y MUESTRA**

### **INICIALIZACIÓN DE LA POBLACIÓN**

La población de cadenas binarias, que representan a los parámetros, se genera aleatoriamente con una distribución uniforme, sobre el espacio de búsqueda, a partir del conocimiento del número de bits de cada parámetro, el ancho de la cadena y el tamaño de la población. El tamaño de la población está en función de la complejidad del problema a resolver, del ancho de las cadenas de la población y de los recursos de procesamiento. Con un tamaño pequeño existe poca probabilidad de encontrar la solución y con poblaciones muy grandes se utiliza mucho tiempo de proceso.

**Ejemplo 4.2. Inicializar una población de tamaño cinco, considerando los datos del Ejemplo 4.1.**

Dado que la población es creada aleatoriamente, se podría tener una población como la siguiente:

| Población Inicial | Número de Cadena |
|-------------------|------------------|
| 1 0 1 0 0 1 0     | Cadena 1         |
| 0 0 1 1 1 0 1     | Cadena 2         |
| 0 1 1 0 1 1 0     | Cadena 3         |
| 1 1 1 0 1 0 0     | Cadena 4         |
| 1 1 0 0 1 1 1     | Cadena 5         |

**Tabla 4.3. Inicialización de la población**

Ahora se tienen los datos necesarios para la siguiente fase.

**DECODIFICACIÓN**

El proceso de decodificación convierte cada uno de los individuos de la población (Cadenas binarias), a sus respectivos valores reales, para ser evaluados a través de una función objetivo. A los individuos representados por sus valores reales se les conoce como fenotipos. La decodificación se puede realizar por medio del código binario estándar o mediante la aplicación de otros códigos, por ejemplo: el *código grey*, el cual presenta la ventaja principal de que para valores de parámetros solución muy cercanos entre sí, existen pocos cambios que representan a esos números, lo cual se puede tomar como una ventaja sobre el código binario estándar en la solución de muchos problemas de optimización. A continuación se presenta la ecuación 6.10, para el caso binario estándar (ver por ejemplo en [3]).

$$Val_{Par.(i)} = u_i + \frac{v_i - u_i}{2^{l_x - 1}} \left( \sum_{j=0}^{l_x - 1} a_i(l_x - j) 2^j \right)$$

Para el caso de decodificación mediante *código Grey* se tiene:

$$Val_{Par.(i)} = u_i + \frac{v_i - u_i}{2^{l_x - 1}} \left( \sum_{j=0}^{l_x - 1} (\oplus_{k=1}^{l_x - 1} - j) 2^j \right)$$

Donde la relación entre el código binario estándar y el *código Grey* está dada por la siguiente expresión.

$$b_i = \begin{cases} a_i, & \text{si } i = 1 \\ a_{i-1} \oplus a_i, & \text{si } i > 1 \end{cases}$$

Código Binario Estándar a C. Grey

$$a_i = \oplus_{j=1}^i b_j$$

C. Grey a Binario Estándar

Para mostrar la aplicación de esta fase de decodificación, se retoman los resultados del Ejemplo 6.1, utilizando el código binario estándar y mediante la ecuación 6.10, se obtienen los valores reales de los parámetros (fenotipos). Los cuales son mostrados en la columna 3 de La Tabla 6.4.

| Población | Cadena   | Valor Real |
|-----------|----------|------------|
| 1010010   | Cadena 1 | 1.4534     |
| 0011101   | Cadena 2 | -2.7177    |
| 0110110   | Cadena 3 | -0.7502    |
| 1110100   | Cadena 4 | 4.1292     |
| 1100111   | Cadena 5 | 3.1061     |

## **EVALUACIÓN**

Se evalúa la función objetivo tomando como argumentos cada uno de los elementos de la población (parámetros), es decir, los valores reales ya decodificados de cada una de las cadenas binarias. Durante este proceso se asigna una calificación a cada elemento de la población, que indica su grado de aptitud.

## **Reproducción**

La etapa de reproducción está formada por tres fases: Selección, Cruzamiento y mutación. Dentro de esta etapa se puede hacer uso de un parámetro adicional que es conocido como *elitismo*, que permite conservar los mejores elementos a través de las  $n$ 's generaciones [3 y 13].

## **4.3 TECNICAS E INSTRUMENTOS DE RECOLECCION DE DATOS**

### **SELECCIÓN**

Este proceso permite seleccionar, en cada uno de los ciclos evolutivos, a los elementos de la población con mejor valor de aptitud. Los métodos de selección más usados son los siguientes:

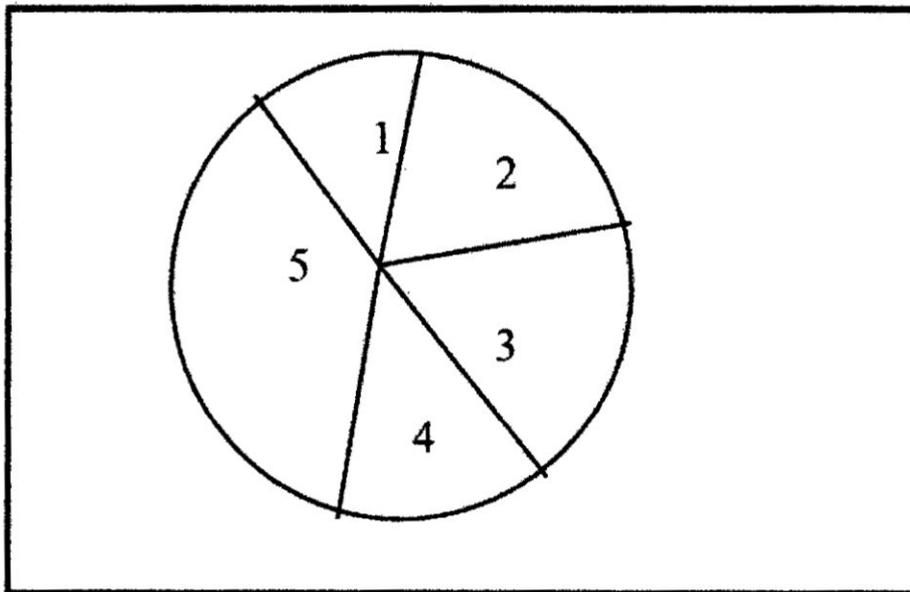
#### **Torneo (tournament)**

Puede resolver problemas de optimización (de minimización o de maximización). Un par de elementos de la población son seleccionados aleatoriamente y el individuo que tenga mejor calificación, es copiado en un lugar de almacenamiento conocido como *mating pool*. Este proceso se repite hasta que este arreglo es llenado con los mejores elementos.

#### **Rueda de la Ruleta (Roulette Wheel)**

Es una de las técnicas de selección más usada. Este mecanismo permite seleccionar de manera aleatoria a los mejores individuos, tomando como base alguna medida de aptitud [19]. Se obtiene la suma ( $F_s$ ) de los valores de aptitud de cada uno

de los individuos de la población actual, donde los individuos son transformados en segmentos contiguos en el intervalo  $[0, F_s]$ . Por ejemplo, la circunferencia de la rueda de la ruleta en La figura 6.8, es la suma de los valores de aptitud de cinco individuos. El Individuo 5 es el que tiene el valor de desempeño más alto y es el que ocupa el intervalo más grande, en contraparte, el individuo 1 es el que tiene el desempeño más pobre y por lo tanto, ocupa el espacio más pequeño de la rueda de la ruleta. Para seleccionar un individuo, primero se genera un número de forma aleatoria, en el intervalo  $[0, F_s]$ , entonces se selecciona un individuo cuyo segmento cubre el número aleatorio. El proceso se repite hasta que se alcanza el número de elementos deseado.

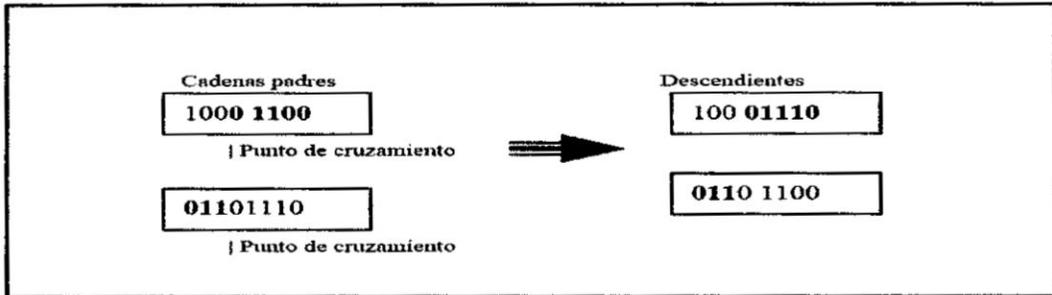


**Figura 4.8. Selección de la Rueda de la Ruleta**

#### **CRUZAMIENTO**

Aunque la rutina de selección podría calificarse también como un operador. Los operadores de cruzamiento y mutación constituyen la herramienta esencial que permite dotar a los descendientes de las mejores características de los padres. Para ilustrar la manera como trabaja el operador de cruzamiento, un mecanismo de cruzamiento de un punto es mostrado en la figura 6.9. Como puede verse, se fija aleatoriamente un punto de cruzamiento. Las porciones de los dos cromosomas que se encuentran a la

derecha del punto de entrecruzamiento son intercambiadas para generar la descendencia. Una tasa funcional ( $p_c$ ) con un valor típico de entre 0.6 y 1.0 es utilizada normalmente como la probabilidad de entrecruzamiento.

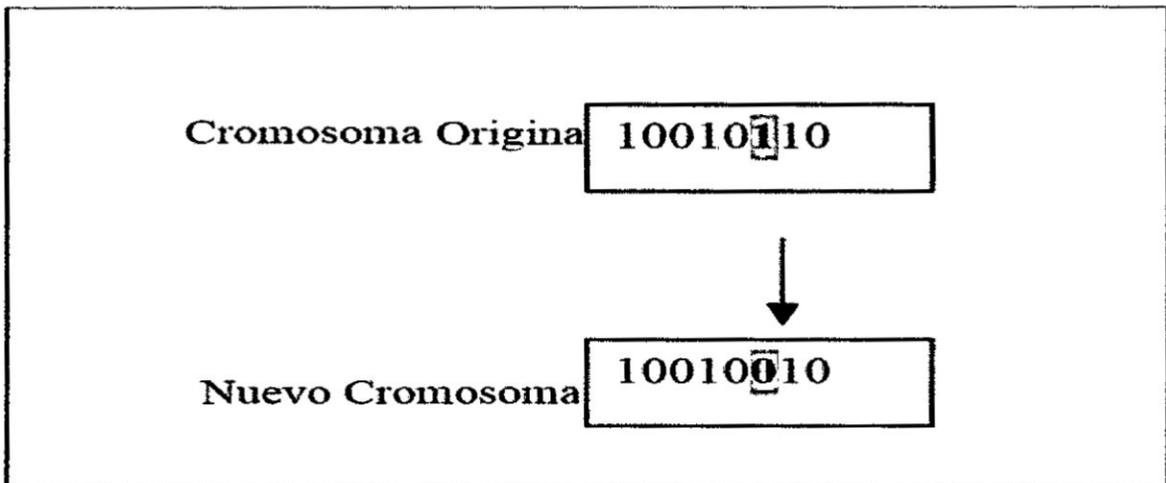


**Figura 4.9. Proceso de cruzamiento**

**MUTACIÓN**

El operador genético de mutación, se aplica a cada descendiente, de forma individual, después del ejercicio de entrecruzamiento. Este proceso altera cada bit aleatoriamente con una pequeña probabilidad ( $p_m$ ) con un valor típico de menos de 0.1. Para ejemplificar este proceso, en la figura 4.10 se muestra a un individuo antes y después de ser mutado.

000



**Figura 4.10. Mutación de un bit**

### 4.3 PLAN DE ANALISIS ESTADISTICO DE DATOS

#### ALGORITMO GENÉTICO SIMPLE

Después de mencionar cada una de las etapas principales de un algoritmo genético simple, conviene presentar de manera esquemática a los bloques que intervienen en el proceso genético, así como el orden en la ejecución del mismo. En Figura 4.3 se muestra el Algoritmo genético simple.

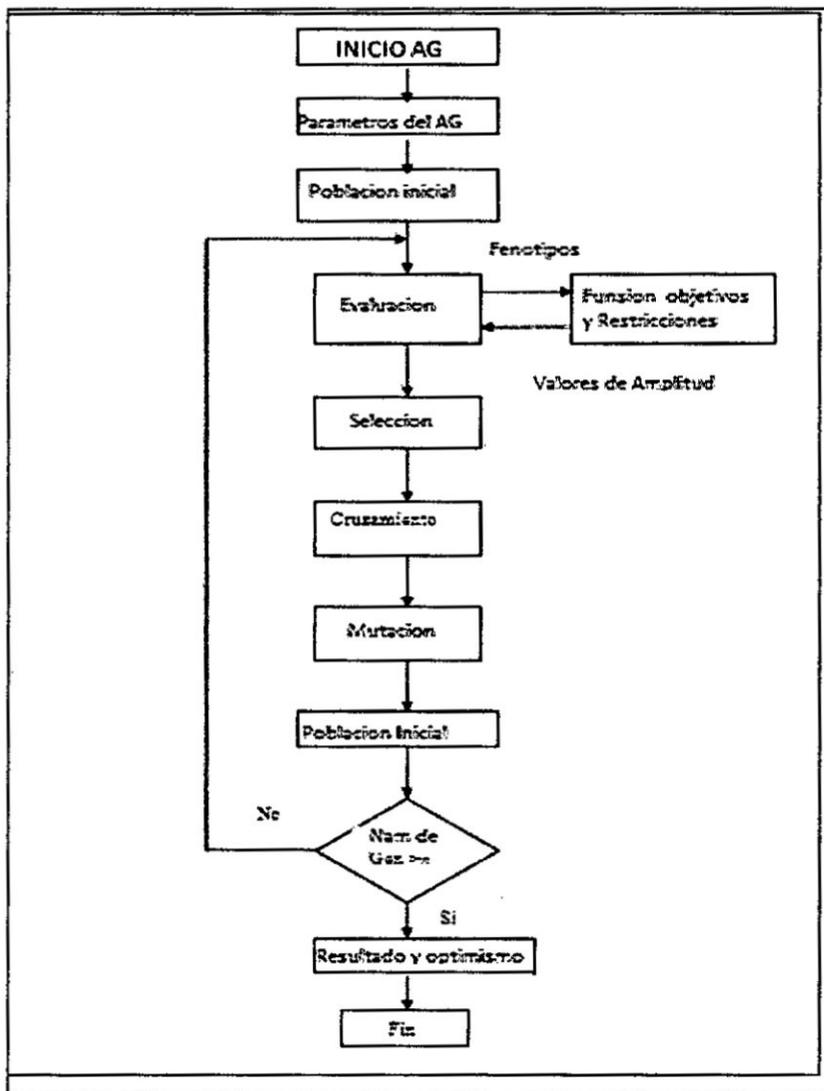


Figura 4.3. Algoritmo Genético Simple

## CAPITULO V

### 5.1 HIPOTESIS DE ESTUDIO

El desempeño del controlador lógico difuso, aplicado al control de posición de un servomotor de C.D., sintonizado con el algoritmo genético simple será mejor o igual que otro CLD sintonizado en base a la experiencia únicamente.

### ***DISEÑO DEL CONTROLADOR DIFUSO***

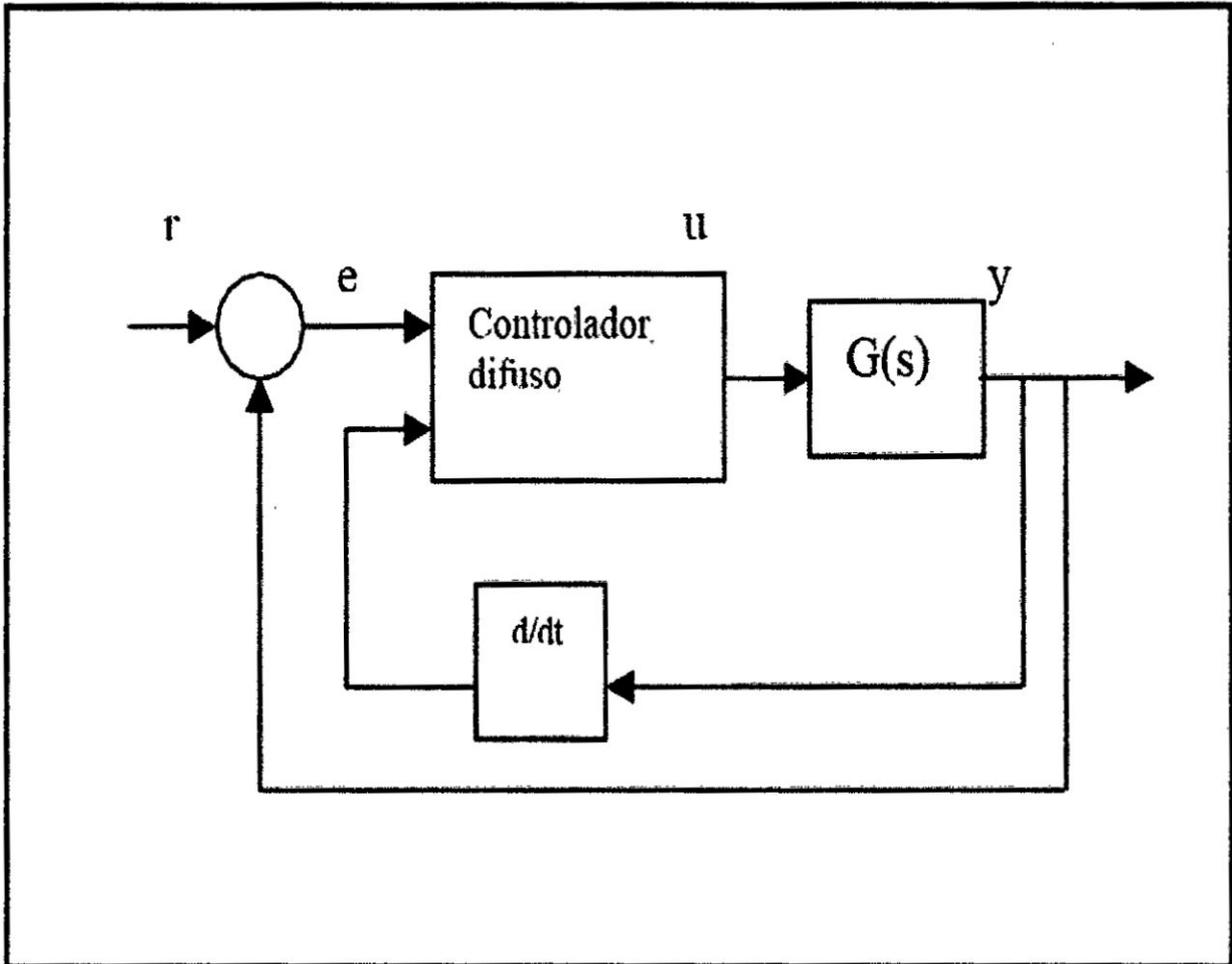
#### **PROBLEMA DE CONTROL**

Se considera el esquema de control mostrado en La figura 8.1, donde:

- $r$  denota la señal de referencia de entrada
- $e$  denota la señal de error de posición
- $u$  denota la señal de control
- $y$  denota la señal de salida
- $G(s)$  denota un sistema lineal e invariante en el tiempo de una entrada-una salida

El problema de control se define a continuación:

*Definición 5.1. Problema de control de seguimiento óptimo (PCSO).* A partir del esquema de control mostrado en la figura 5.1, encontrar un controlador lógico difuso (CLD), que minimice la señal de error  $e$  para una entrada de referencia  $r$ .



**Figura 5.1 Esquema de control de seguimiento**

El modelo del Servomotor experimental fue obtenido con los datos del fabricante, los cuales se presentan en el apéndice de este documento. El modelo de la planta está dado por la ecuación 5.1.

$$G(s) = \frac{65}{s(s + 31)}$$

## MÉTODO DE OPTIMIZACIÓN

El controlador difuso tiene dos señales de entrada y una de salida, las entradas son: La señal de error de posición y variación de la señal de salida (velocidad angular), estas toman siete valores lingüísticos: Negativo Grande (NG), Negativo Medio (NM), Negativo Pequeño (NP), cero (CE), Positivo Pequeño (PP), Positivo Medio (PM) y Positivo Grande (PG). La variable de salida también toma siete valores lingüísticos: Negativo Grande (NG), Negativo Medio (NM), Negativo Pequeño (NP), cero (CE), Positivo Pequeño (PP), Positivo Medio (PM) y Positivo Grande (PG). El controlador difuso se diseña con los conjuntos difusos presentados en Las Figuras 5.2, 5.3 y 5.4, en base a los valores lingüísticos definidos.

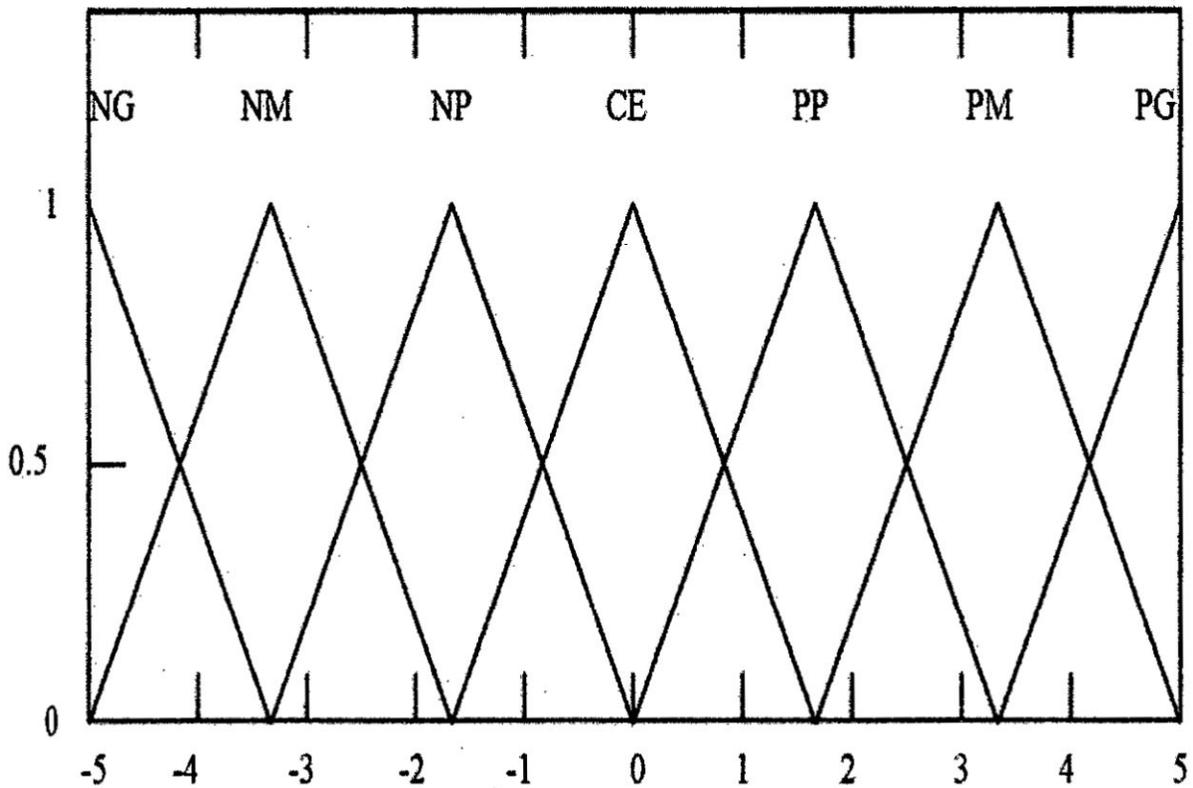
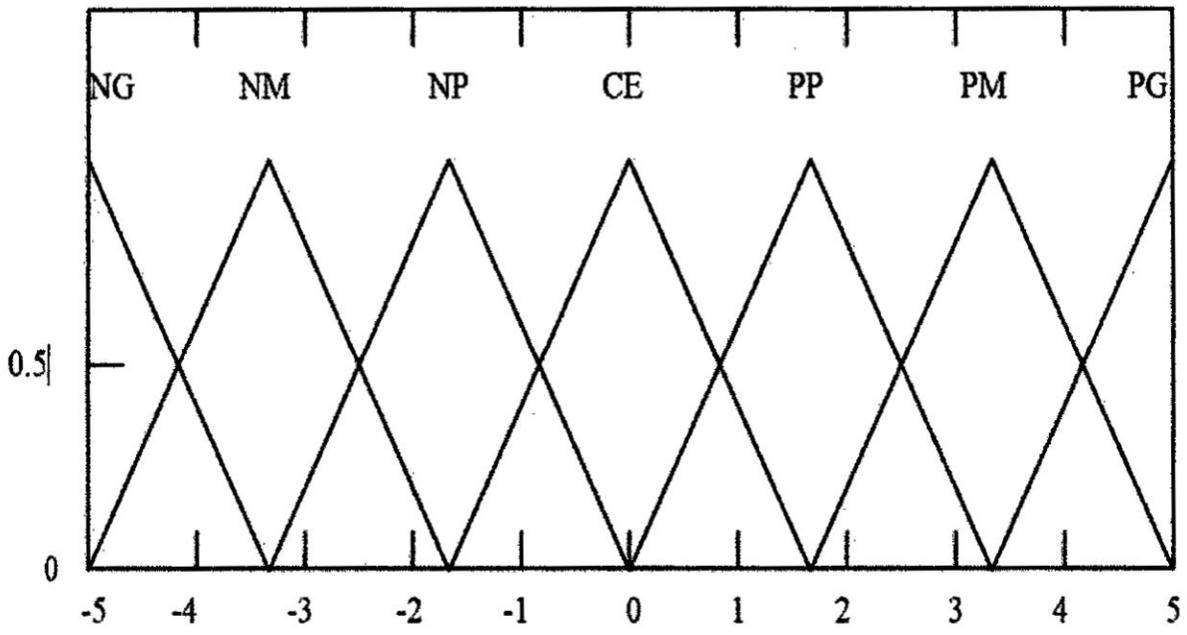
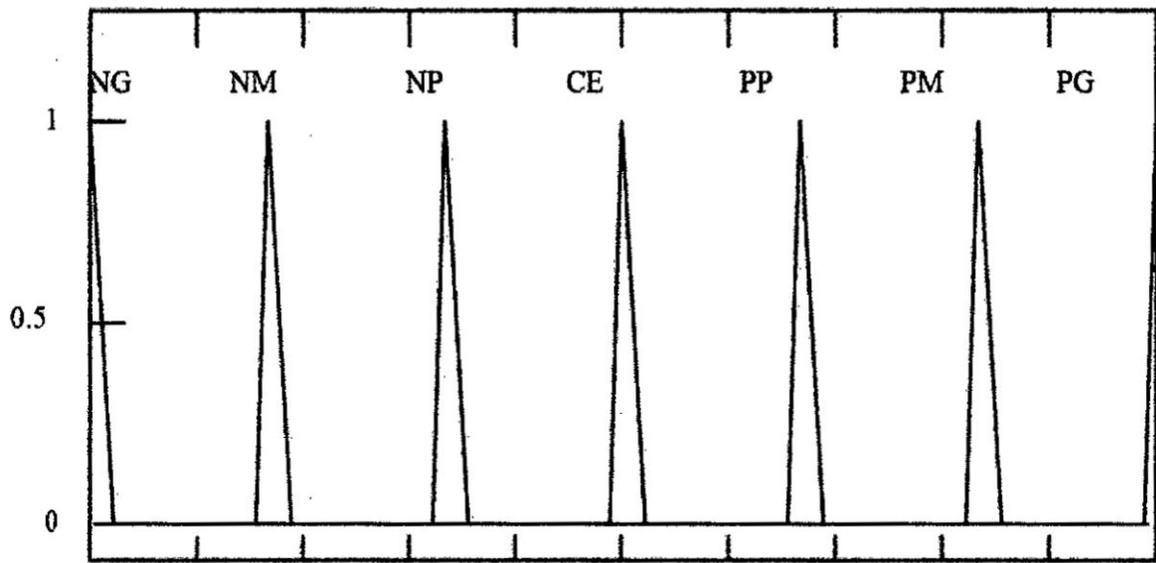


Figura 5.2. Conjunto difuso de entrada: Error de posición



**Figura 5.3 Conjunto difuso de entrada: Cambio de error**



**Figura 5.4 Conjunto difuso de salida: Salida de control**

La función objetivo propuesta en esta tesis es el criterio de la Integral del Valor Absoluto, *IAE* (por sus iniciales en inglés), la cual está definida en la ecuación 5.2, [11 y 15].

$$I_1 = \int_0^{\infty} |e(t)| dt$$

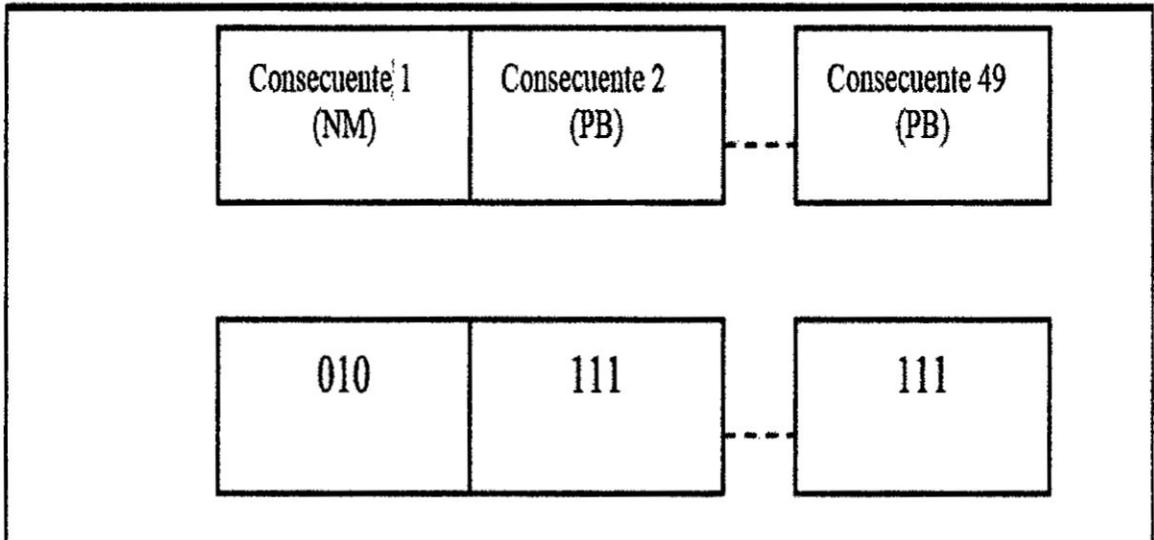
La función objetivo dada por la ecuación 5.2, es minimizada durante el proceso de optimización. El Criterio *IAE* se usa cuando se desea una respuesta buena a los cambios de la señal de referencia. Así los valores de interés son: tiempo de crecimiento, máximo sobrepaso, tiempo de máximo sobrepaso, relación de decaimiento y tiempo de establecimiento.

Por simplicidad, se usa el código binario, por lo que los valores: {Negativo Grande (NG), Negativo Medio (NM), Negativo Pequeño (NP), cero (CE), Positivo Pequeño (PP), Positivo Medio (PM) y Positivo Grande (PG)}, de la variable de control, se codifica como: 001, 010, 011, 100, 101, 110, y 111. En La Tabla 8.1 se presenta un ejemplo de un controlador difuso en particular, para el cual el cromosoma correspondiente sería: {100 110 111 111 101 101 101 101}...{001 001 100}.

**Tabla Reglas de control difuso**

| ec<br>u<br>e | PB | PM | PS | ZO | NS | NM | NB |
|--------------|----|----|----|----|----|----|----|
| PB           | ZO | PS | ZO | NS | ZE | ZE | NM |
| PB           | PM | PB | PM | ZO | PS | ZE | NS |
| PS           | PB | PB | PB | ZO | PS | NS | NS |
| ZO           | PB | PB | PS | ZO | NS | NS | NB |
| NS           | PS | ZE | ZE | PS | NS | NS | NB |
| NM           | PM | ZE | ZE | PS | NM | NB | NB |
| NB           | PM | ZE | ZE | PN | NG | NG | ZO |

Cada cromosoma tiene 147 números binarios, que corresponden a los 49 consecuentes de un controlador difuso en particular. El algoritmo genético opera con una población de individuos, en la cual cada uno de estos (cromosoma), contiene los consecuentes de la base de reglas difusas. En parte inferior de La figura 8.5 se muestra un ejemplo de un cromosoma codificado.



**Figura 5.5 Cromosoma Codificado**

Durante la evaluación de los individuos, el valor de aptitud de cada uno de ellos, se obtiene minimizando el criterio *IAE* (función objetivo), este valor se obtiene a través de una simulación en **MATLAB<sup>®</sup>**, por cada uno de los individuos de la población.

## PROCESO GENÉTICO

El proceso genético inicia con la creación de una población de individuos (creada aleatoriamente), donde cada individuo contiene a los 49 consecuentes de un controlador difuso en particular. Sin embargo para que los 49 consecuentes puedan ser usados en una función de **MATLAB<sup>®</sup>**, para formar la base de reglas, se debe convertir

al individuo, codificado como una cadena binaria, a números enteros, en el rango de 1 a 7, donde: Negativo Grande (NG)=1, Negativo Medio (NM)=2, Negativo Pequeño (NP)=3, cero (CE)=4, Positivo Pequeño (PP)=5, Positivo Medio (PM)=6 y Positivo Grande (PG)=7. En el arreglo presentado por la Matriz 8.3 se muestran los 49 consecuentes representados por  $c_n$ .

$$\begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \\ c_8 & c_9 & c_{10} & c_{11} & c_{12} & c_{13} & c_{14} \\ c_{15} & c_{16} & c_{17} & c_{18} & c_{19} & c_{20} & c_{21} \\ c_{22} & c_{23} & c_{24} & c_{25} & c_{26} & c_{27} & c_{28} \\ c_{29} & c_{30} & c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \\ c_{36} & c_{37} & c_{38} & c_{39} & c_{40} & c_{41} & c_{42} \\ c_{43} & c_{44} & c_{45} & c_{46} & c_{47} & c_{48} & c_{49} \end{bmatrix}$$

Para formar la base de reglas de un controlador difuso, se requieren los valores de los antecedentes, en este caso, Los Antecedentes 1 y 2, que se encuentran fijos en una matriz dentro de la función de MATLAB<sup>®</sup>, como se muestra en la Matriz 8.4.

$$\begin{bmatrix} a1_1 a2_1 c_1 & a1_2 a2_2 c_2 & \dots & a1_7 a2_7 c_7 \\ a1_8 a2_8 c_8 & a1_9 a2_9 c_9 & \dots & a1_{14} a2_{14} c_{14} \\ a1_{15} a2_{15} c_{15} & a1_{16} a2_{16} c_{16} & \dots & a1_{22} a2_{22} c_{22} \\ \vdots & \vdots & \ddots & \vdots \\ a1_{43} a2_{43} c_{43} & a1_{44} a2_{44} c_{44} & \dots & a1_{49} a2_{49} c_{49} \end{bmatrix}$$

Donde:

a1: denota Al Antecedente 1

a2: denota Al Antecedente 2

Cn: denota Al Consecuente n

Los valores lingüísticos de los antecedentes, corresponden a los valores enteros que se presentan a continuación:

Negativo Grande (NG)=1

Negativo Medio (NM)=2

Negativo Pequeño (NP)=3

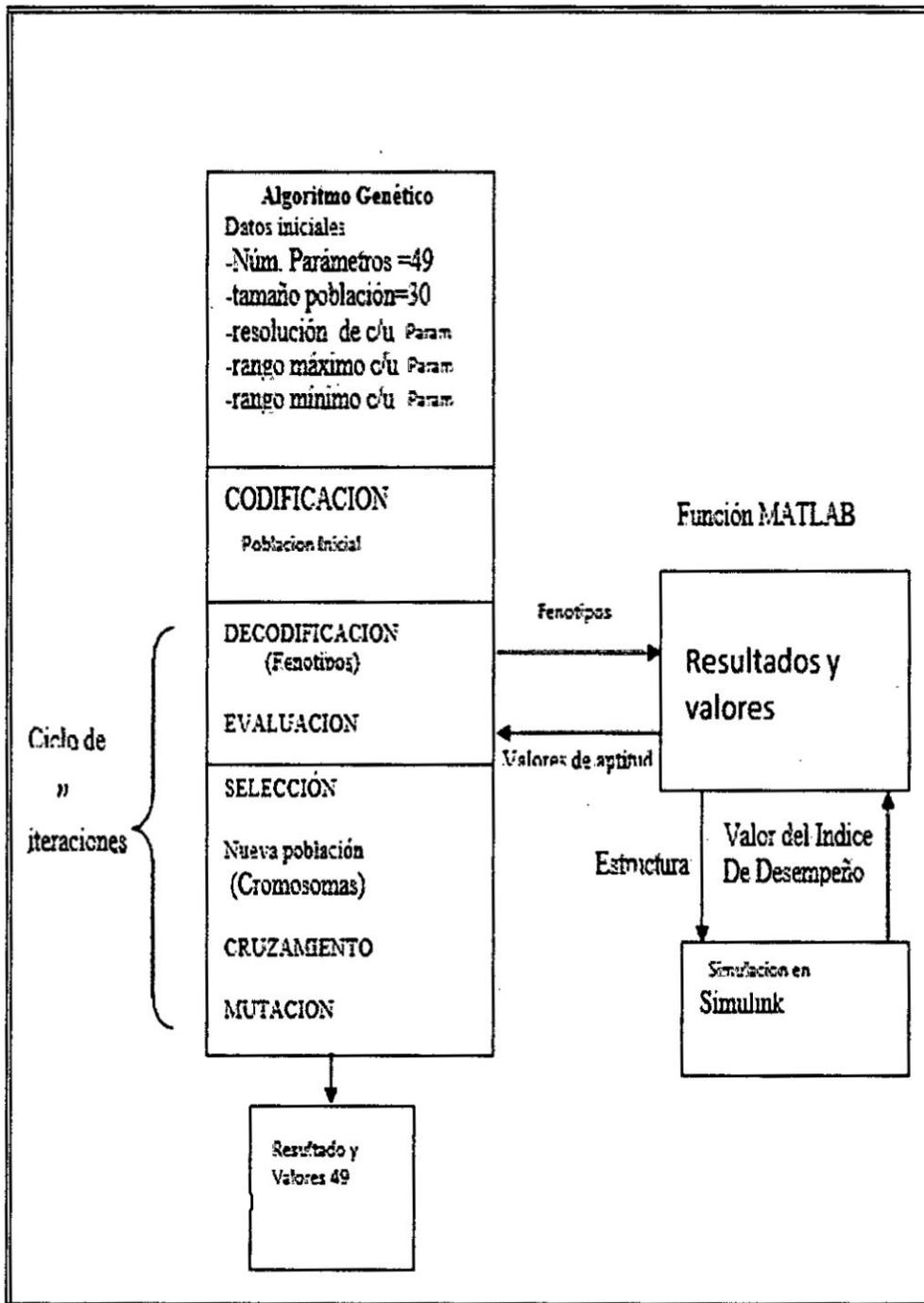
Cero (CE) =4

Positivo Pequeño (PP)=5

Positivo Medio (PM)= 6 Positivo Grande (PG)= 7

Después de que se tienen todos los datos de la matriz 8.4, la función de MATLAB<sup>®</sup>  $a=addrule(a,ruleList)$ , genera la base de reglas difusas del controlador [8], donde ruleLista, corresponde a la matriz 8.4.

Finalmente, junto con la base de reglas, los rangos de las variables de entrada-salida, los datos de las funciones de membresía: la Función de MATLAB<sup>®</sup>, tiene como salida un archivo MATLAB<sup>®</sup> FIS (Fuzzy Inference System), que corresponde a una estructura donde se tienen todos los datos de inferencia difusa del sistema. Esta estructura FIS es utilizada en la simulación por el esquema de control difuso. Después de la simulación se obtiene el valor del índice de desempeño, el cual es enviado al algoritmo genético, para continuar con el proceso. Para mayor claridad, el proceso completo se presenta en la figura 5.6.



**Figura 5.6 Proceso de optimización**

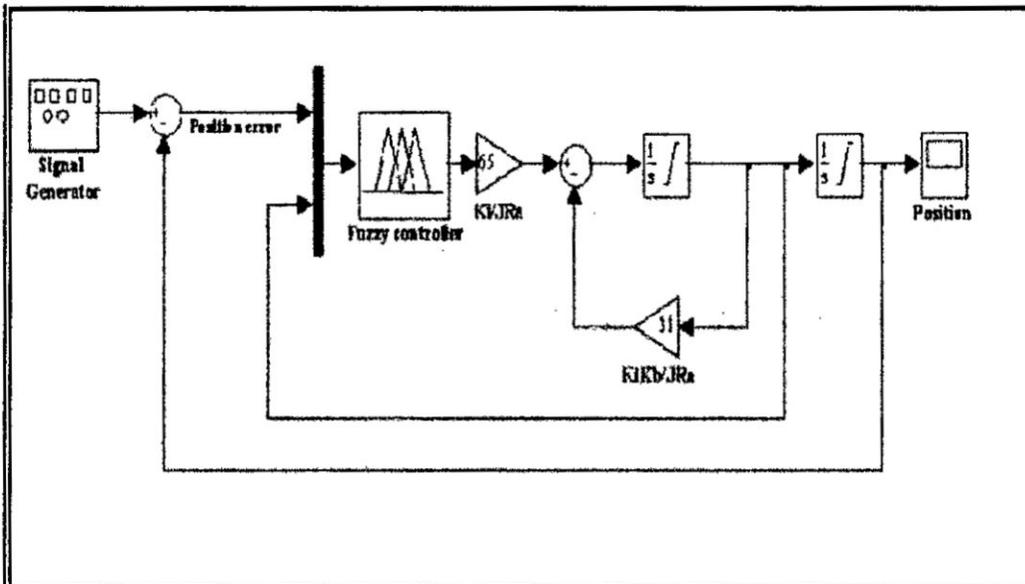
Después de aplicar el procedimiento descrito en este capítulo, se obtuvo la base de reglas que se presenta en la Tabla

| Reglas Difusas |   |
|----------------|---|
| 1              | If (error de posición es PG) and (cambio de error es NG) then (salida de control es PP) |
| 2              | If (error de posición es PP) and (cambio de error es PM) then (salida de control es PG) |
| 3              | If (error de posición es PG) and (cambio de error es PM) then (salida de control es PG) |
| 4              | If (error de posición es PG) and (cambio de error es PG) then (salida de control es PG) |
| 5              | If (error de posición es PG) and (cambio de error es PG) then (salida de control es PG) |
| 6              | If (error de posición es PG) and (cambio de error es NM) then (salida de control es NG) |
| 7              | If (error de posición es PG) and (cambio de error es NG) then (salida de control es PM) |
| 8              | If (error de posición es PM) and (cambio de error es CE) then (salida de control es PP) |
| 9              | If (error de posición es PM) and (cambio de error es PP) then (salida de control es PG) |
| 10             | If (error de posición es PM) and (cambio de error es PM) then (salida de control es NG) |
| 11             | If (error de posición es PM) and (cambio de error es PG) then (salida de control es PP) |
| 12             | If (error de posición es PM) and (cambio de error es NP) then (salida de control es PG) |
| 13             | If (error de posición es PM) and (cambio de error es NM) then (salida de control es PG) |
| 14             | If (error de posición es PM) and (cambio de error es NG) then (salida de control es PG) |
| 15             | If (error de posición es PP) and (cambio de error es NG) then (salida de control es PM) |
| 16             | If (error de posición es PP) and (cambio de error es NM) then (salida de control es PG) |
| 17             | If (error de posición es PP) and (cambio de error es NP) then (salida de control es NG) |
| 18             | If (error de posición es PP) and (cambio de error es CE) then (salida de control es PG) |
| 19             | If (error de posición es PP) and (cambio de error es PP) then (salida de control es PG) |
| 20             | If (error de posición es PP) and (cambio de error es PM) then (salida de control es PG) |
| 21             | If (error de posición es PP) and (cambio de error es PG) then (salida de control es NG) |
| 22             | If (error de posición es CE) and (cambio de error es NG) then (salida de control es PP) |
| 23             | If (error de posición es CE) and (cambio de error es NM) then (salida de control es PM) |
| 24             | If (error de posición es CE) and (cambio de error es NP) then (salida de control es PG) |
| 25             | If (error de posición es CE) and (cambio de error es CE) then (salida de control es NG) |
| 26             | If (error de posición es CE) and (cambio de error es PP) then (salida de control es NG) |
| 27             | If (error de posición es CE) and (cambio de error es PM) then (salida de control es NG) |
| 28             | If (error de posición es CE) and (cambio de error es PG) then (salida de control es CE) |
| 29             | If (error de posición es NP) and (cambio de error es NG) then (salida de control es CE) |
| 30             | If (error de posición es NP) and (cambio de error es NM) then (salida de control es NG) |
| 31             | If (error de posición es NP) and (cambio de error es NP) then (salida de control es NM) |
| 32             | If (error de posición es NP) and (cambio de error es CE) then (salida de control es NP) |
| 33             | If (error de posición es NP) and (cambio de error es PP) then (salida de control es NM) |
| 34             | If (error de posición es NP) and (cambio de error es PM) then (salida de control es CE) |

|    |   |
|----|---|
| 35 | If (error de posición es NP) and (cambio de error es PG) then (salida de control es NG) |
| 36 | If (error de posición es NM) and (cambio de error es NG) then (salida de control es NG) |
| 37 | If (error de posición es NM) and (cambio de error es NM) then (salida de control es NM) |
| 38 | If (error de posición es NM) and (cambio de error es NP) then (salida de control es NP) |
| 39 | If (error de posición es NM) and (cambio de error es CE) then (salida de control es NG) |
| 40 | If (error de posición es NM) and (cambio de error es PP) then (salida de control es NG) |
| 41 | If (error de posición es NM) and (cambio de error es PM) then (salida de control es NM) |
| 42 | If (error de posición es NM) and (cambio de error es PG) then (salida de control es CE) |
| 43 | If (error de posición es NG) and (cambio de error es NG) then (salida de control es CE) |
| 44 | If (error de posición es NG) and (cambio de error es NM) then (salida de control es CE) |
| 45 | If (error de posición es NG) and (cambio de error es NP) then (salida de control es CE) |
| 46 | If (error de posición es NG) and (cambio de error es CE) then (salida de control es CE) |
| 47 | If (error de posición es NG) and (cambio de error es PP) then (salida de control es CE) |
| 48 | If (error de posición es NG) and (cambio de error es PM) then (salida de control es CE) |
| 49 | If (error de posición es NG) and (cambio de error es PG) then (salida de control es NG) |

**Tabla Base de reglas obtenidas mediante el método de optimización**

En la figura 5.7 se presenta el diagrama esquemático que se usó para la simulación de cada uno de los controladores difusos en el proceso de optimización usando un algoritmo genético simple.

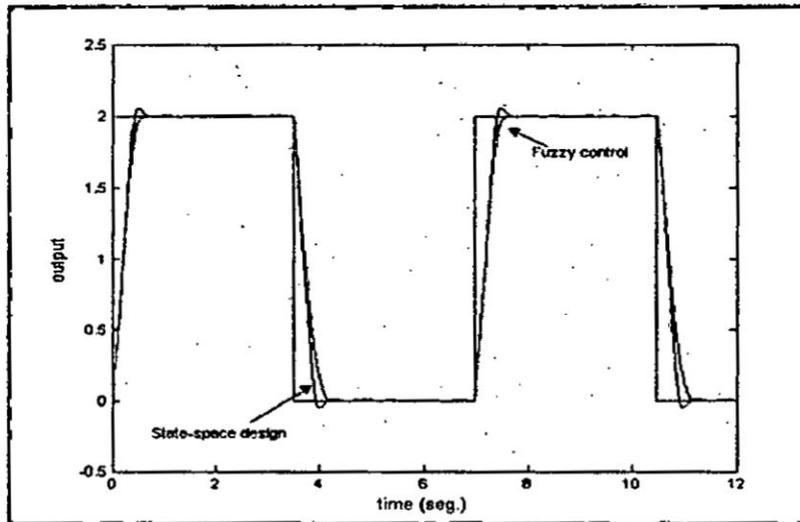


**Figura 5.7 Control de posición: Esquema en Simulink®**

Por medio del método propuesto se obtuvo la matriz de datos (5.5), que contiene los 49 consecuentes del controlador difuso diseñado.1j

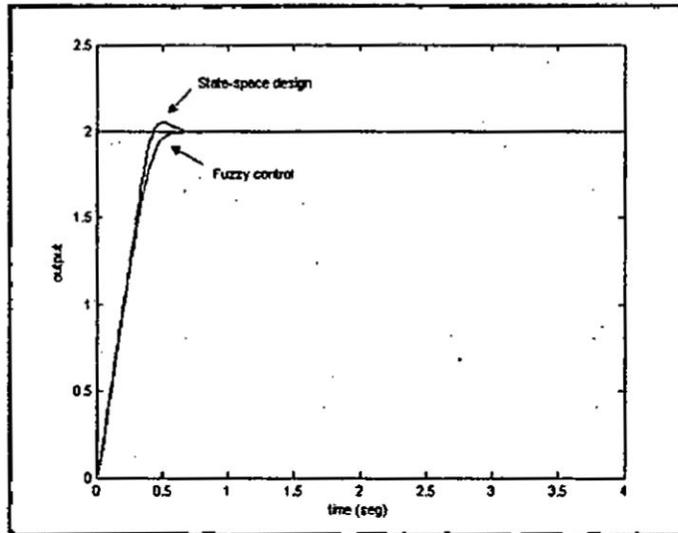
|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 5 | 7 | 7 | 7 | 5 | 4 | 6 |
| 5 | 7 | 4 | 5 | 7 | 7 | 7 |
| 6 | 7 | 4 | 7 | 7 | 7 | 4 |
| 5 | 6 | 7 | 4 | 4 | 4 | 1 |
| 1 | 4 | 2 | 3 | 2 | 1 | 4 |
| 4 | 2 | 3 | 4 | 4 | 2 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 4 |

Las curvas que se muestran en la figura 5.8, corresponden a la simulación del controlador difuso diseñado en esta investigación. Con la finalidad de tener una referencia acerca del desempeño del controlador difuso, en la figura 8.8 se muestra también, la salida del sistema cuando se utiliza un controlador basado en la realimentación de estados [4], cuyo vector de ganancia es:  $k=[4 \ -0.016]$ .



**Figura 5.8 Respuesta temporal del sistema: Señal cuadrada**

En La figura 5.9 se puede apreciar la respuesta temporal del sistema ante una entrada de referencia escalón, donde se aprecia que la señal de salida alcanza el valor de la referencia sin sobre paso.

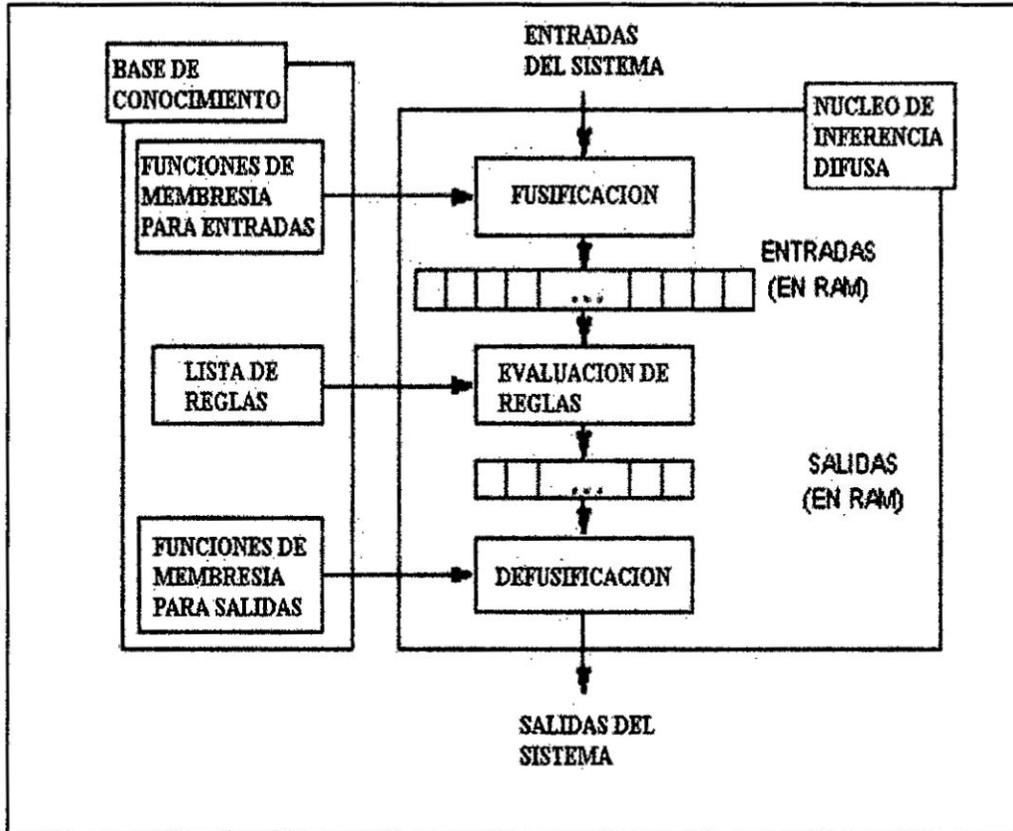


**Figura 5.9 Respuesta temporal del sistema: Señal escalón**

### **PROCEDIMIENTO PARA IMPLANTAR EL CONTROLADOR LÓGICO DIFUSO**

El Controlador difuso se implantará en una tarjeta de evaluación basada en el microcontrolador HCS12 de MOTOROLA, debido a que este microcontrolador, cuenta con las funciones específicas para desarrollar el proceso de inferencia difusa.

El control difuso se compone de dos partes. La primera parte se conoce como base de conocimiento, formada por las funciones de membresía de entrada, base de reglas y funciones de membresía de salida. La segunda parte es un programa que realiza el problema de control, el cual se puede dividir en fusificación, evaluación de reglas y defusificación. El proceso de control difuso se presenta en La figura 5.10.



**Figura 5.10 Diagrama de Bloques de un Sistema de Lógica Difusa**

Una vez desarrollada la base de conocimiento, ya sea en base a la experiencia o por el método de optimización, ésta, necesita ser organizada como una base de datos dentro de una sección de memoria, para luego ser utilizada por el núcleo de inferencia del microcontrolador. Pero a su vez esta base de conocimiento tiene que cumplir, para el caso del microcontrolador MC9s12E, con un formato preestablecido.

## **BASE DE CONOCIMIENTO**

### **Funciones de membresía de entrada**

El manual de referencia de MOTOROLA<sup>®</sup> especifica cómo debe ser la estructura de una función de membresía. En La figura 8.11 se observa esta consideración. En este caso la representación seleccionada fue la forma triangular, por

lo tanto podemos reconocer 4 características de esta representación: los 2 puntos que delimitan la base y las 2 pendientes que conforman los lados del triángulo; con estos datos se crea la primera parte de la base de conocimiento.

A partir de la primera etiqueta, se comienza a reservar área de memoria, la primera dirección es ocupada por el primer punto de la función de pertenencia (visto de izquierda a derecha), la siguiente dirección: el segundo punto, la siguiente dirección: la primera pendiente y la última dirección: la segunda pendiente.

Este es el mismo procedimiento para cada una de las etiquetas lingüísticas, por lo que las funciones de membresía, requieren un total de cuatro bytes de memoria por cada etiqueta.

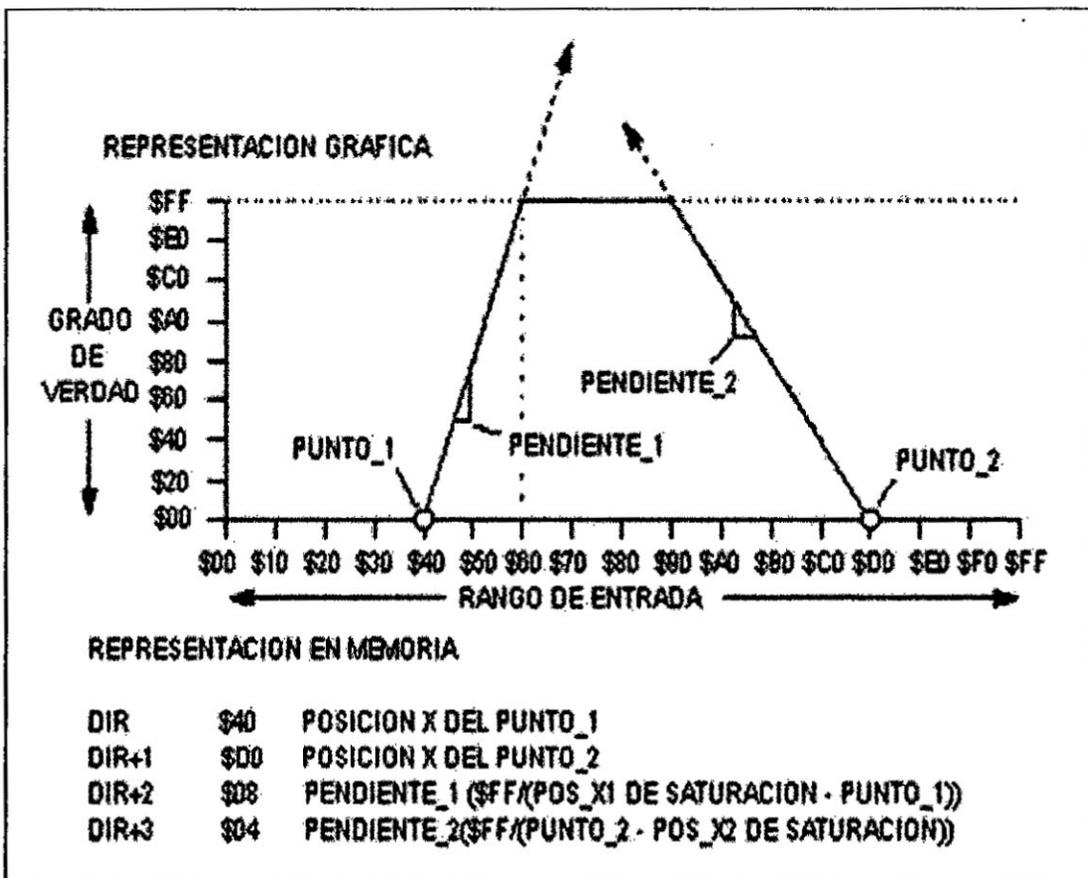
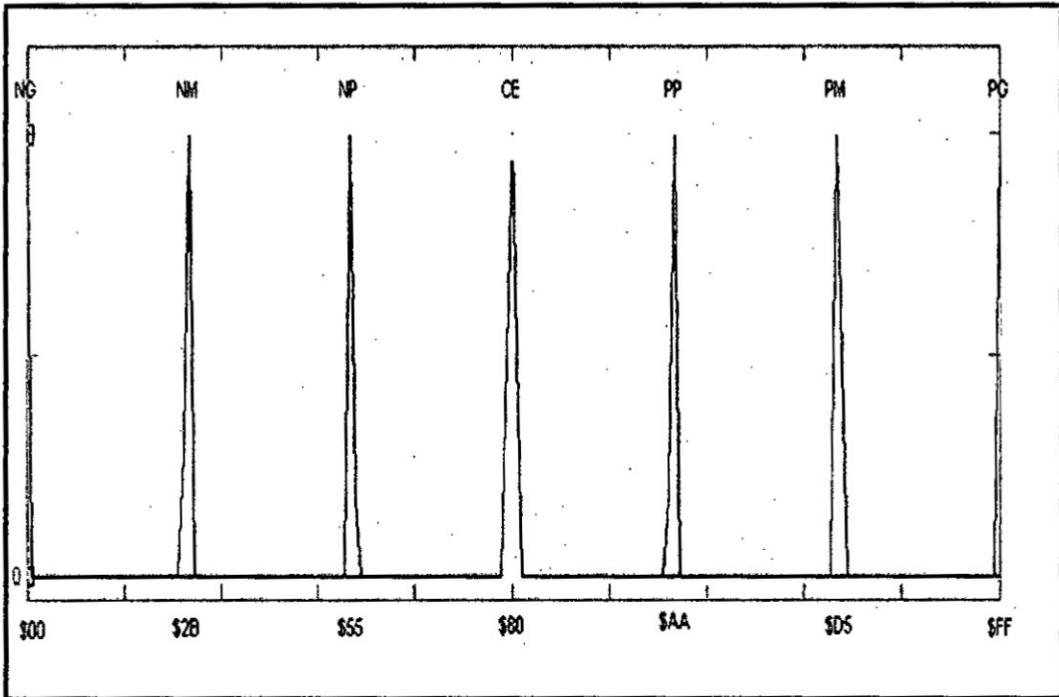


Figura 5.11 Método de diseño de las funciones de membresía en el microcontrolador

### **Función de membresía de salida**

Esta función de salida es de tipo singleton (ver Figura 5.12), por lo que cada etiqueta ocupa un byte de memoria.



**Figura 5.12 Funciones de membresía de salida**

### **Reglas de inferencia**

Para poder definir las reglas se utiliza un método de organización, que consiste en señalar el fin de cada una de las reglas, con un valor reservado; para los antecedentes y consecuentes también se hace uso de un separador. Cuando se está ejecutando el programa, este separador afecta a uno de los bits del registro de condición del microcontrolador (bit V), lo cual establece la diferencia entre un antecedente o un consecuente o, si ha terminado la evaluación de reglas. En la figura 5.12 se presenta un ejemplo de la codificación de una regla difusa.

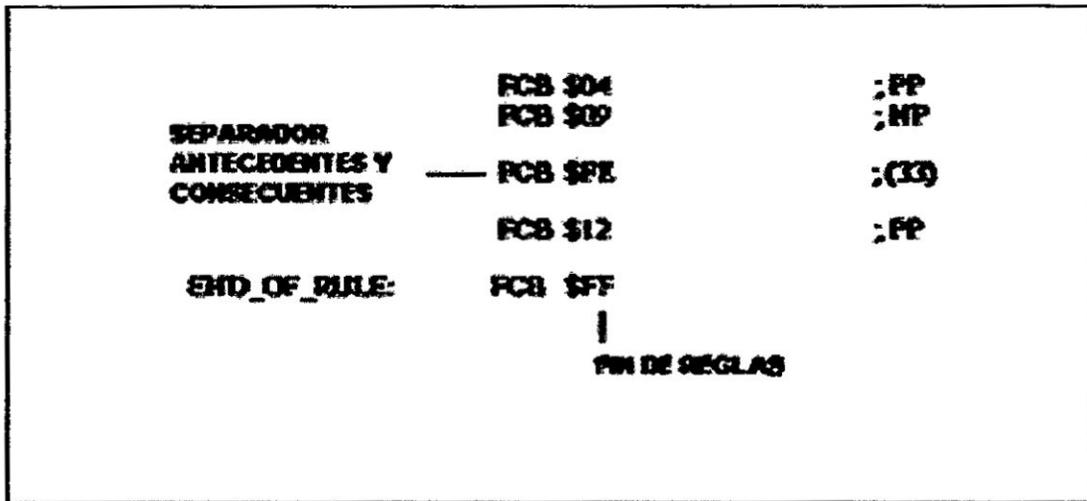


Figura 5.13 Ejemplo de declaración de reglas

También, es necesario codificar cada uno de los componentes de las reglas de inferencia, como se muestra en la Tabla .

| ANTECEDENTE 1 |      | ANTECEDENTE 2 |      | CONSECUENTE |      |
|---------------|------|---------------|------|-------------|------|
| NG            | \$00 | NG            | \$07 | NG          | \$0E |
| NM            | \$01 | NM            | \$08 | NM          | \$0F |
| NP            | \$02 | NP            | \$09 | NP          | \$10 |
| CE            | \$03 | CE            | \$0A | CE          | \$11 |
| PP            | \$04 | PP            | \$0B | PP          | \$12 |
| PM            | \$05 | PM            | \$0C | PM          | \$13 |
| PG            | \$06 | PG            | \$0D | PG          | \$14 |

Tabla Codificación de las reglas difusas (para HC12)

### NÚCLEO DE INFERENCIA

En la figura 8.14, se muestra el núcleo de inferencia utilizado para el CLD. En los comentarios se observa el procedimiento de cada una de las líneas del código. Para que el código de inferencia difusa se ejecute, solo se necesitan dos valores de entrada, que en este caso son *VALOR\_ACT* y *VALOR\_ACT+1* y la salida se obtiene en una

variable llamada SAL\_COG.

|    |        |           |      |             |                                      |
|----|--------|-----------|------|-------------|--------------------------------------|
| 01 | [2]    | FUZZYFICA | LDX  | #FMS_ENT    | : Apunta a definiciones de FM        |
| 02 | [2]    |           | LDY  | #ENT_FUZ    | : Apunta a tabla de entradas fuzzy   |
| 03 | [3]    |           | LDAA | VALOR_ACT   | : Trae 1er valor de entrada          |
| 04 | [1]    |           | LDAB | #7          | : 7 etiquetas por entrada            |
| 05 | [5]    | LAZO1     | MEM  |             | : Evalúa una FM                      |
| 06 | [3]    |           | DBNE | B,LAZO1     | : Para 7 etiquetas por entrada       |
| 07 | [3]    |           | LDAA | VALOR_ACT+1 | : Trae 2do valor de entrada          |
| 08 | [1]    |           | LDAB | #7          | : 7 etiquetas por entrada            |
| 09 | [5]    | LAZO2     | MEM  |             | : Evalúa una FM                      |
| 10 | [3]    |           | DBNE | B,LAZO2     | : Para 7 etiquetas por entrada       |
| 11 | [1]    |           | LDAB | #7          | : contador del lazo                  |
| 12 | [2]    | EVALUA    | CLR  | 1,Y+        | : Limpia sal. Fuzzy e incr. Apunt.   |
| 13 | [3]    |           | DBNE | B,EVALUA    | : hasta limpiar todas las sal. Fuzzy |
| 14 | [2]    |           | LDX  | #REGLA1     | : Apunta a 1er elemento de reglas    |
| 15 | [2]    |           | LDY  | #ENT_FUZ    | : Apunta a ent y sal. fuzzy          |
| 16 | [1]    |           | LDAA | #SFF        | : Inicia A (y limpia bit V)          |
| 17 | [3n+4] |           | REV  |             | : Procesa lista de reglas            |
| 18 | [2]    | DEFUZ     | LDY  | #SAL_FUZ    | : Apunta a salidas fuzzy             |
| 19 | [1]    |           | LDX  | #POS_SGLTN  | : Apunta a posiciones singleton      |
| 20 | [1]    |           | LDAB | #7          | : 7 salidas fuzzy por salida crisp   |
| 21 | [8b+0] |           | WAV  |             | : Hace suma por prom ponderado       |
| 22 | [11]   |           | EDIV |             | : Divide por suma ponderada          |
| 23 | [1]    |           | TFR  | Y, D        | : Pone resultado en A:B              |
| 24 | [3]    |           | STAB | SAL_COG     | : Almacena salida del sistema        |

Figura 5.14 Núcleo de inferencia difuso

## CRITERIO BASADO EN LA EXPERIENCIA PARA SELECCIONAR LAS REGLAS DEL CLD

La estrategia de control se combina principalmente de las siguientes sentencias:

1. Si error es PG y cambio de error es alguno entonces salida de control es PG
2. Si error es PP y cambio de error es PP ó CE entonces salida de control es PP
3. Si error es CE y cambio de error es PP entonces salida de control es CE
4. Si error es CE y cambio de error es NP entonces salida de control es NP
5. Si error es NP y cambio de error es NP entonces salida de control es NP
6. Si error es NG y cambio de error es alguno entonces salida de control es NG

La Sentencia 1 implica una condición general cuando la posición actual con respecto a la referencia es grande. De esta manera se requiere una salida grande para mover el eje del motor y alcanzar la salida deseada rápidamente. La regla 2 se aplica cuando el error empieza a disminuir y el motor se aproxima a la posición deseada. En este caso se requiere una salida pequeña. La sentencia 3 presenta el caso cuando el error se está aproximando a cero. Sin embargo, hay que considerar la inercia del motor

por lo que la salida debe ser cero para evitar el sobrepaso al mínimo. Así mismo, la regla 4, se aplica cuando ocurre un sobrepaso. Entonces se aplica una salida pequeña en sentido inverso, para regresar el eje del motor. La Sentencia 6 implica una condición inversa de la regla 1.

## **5.2 RESULTADOS**

Los resultados experimentales se obtuvieron por medio de un osciloscopio digital, el cual permite almacenar los datos, en una unidad de memoria y posteriormente, fueron graficados usando MATLAB<sup>®</sup>. Los resultados corresponden a las respuestas de: Un sistema de control de posición de un servomotor mediante un controlador lógico difuso (CLD), sintonizado mediante la experiencia adquirida y CLD sintonizado con un algoritmo genético simple, lo cual permite comparar dichos resultados y demostrar que una sintonización en base a los algoritmos genéticos proporciona un mejor desempeño. Ambos resultados son producto de las pruebas experimentales al sistema mediante un microcontrolador con un núcleo de inferencia de lógica difusa (MC9s12E128) y todas las mediciones fueron hechas directamente del sistema y trasladadas a la plataforma gráfica de MATLAB<sup>®</sup>.

### **RESULTADOS DEL CLD SINTONIZADO EN BASE A LA EXPERIENCIA**

Esta primera base de conocimiento fue desarrollada en base a la experiencia, por lo que el número de reglas fue definido analizando detalladamente el funcionamiento del sistema de control de posición. En la tabla 9.1 se presenta la base de reglas del controlador difuso, el cual está formado por 32 reglas de inferencia; la acción de control se forma en la intersección de las filas, que corresponden al error de posición, y las columnas que corresponden a la velocidad angular. Como ejemplo, se puede enunciar a continuación una de las reglas difusas, la cual se forma con las posiciones sombreadas de la Tabla 5.2.1:

Si el error de posición es PG y la velocidad es PG, entonces la acción de control es PM

| VELOCIDAD |    | PG | PM | PP | CE | NP | NM | NG |
|-----------|----|----|----|----|----|----|----|----|
| POSICION  | PG | PM | PG |    | PG |    |    |    |
|           | PM | PM | PG | PG | PG |    |    |    |
|           | PP | PM | PG | PG | PG | PP |    |    |
|           | CE | NP | NP | CE | CE | CE | PP | PP |
|           | NP |    |    | CE | NG | NG | NG | NP |
|           | NM |    |    |    | NG | NG | NM | NP |
|           | NG |    |    |    | NG | NG | NG |    |

Tabla 5.2.1 Matriz de reglas de control en base a la experiencia

En La figura 5.2.1 se presenta la señal de salida del sistema de control de posición, que corresponde a la parte de subida de la señal cuadrada de referencia.

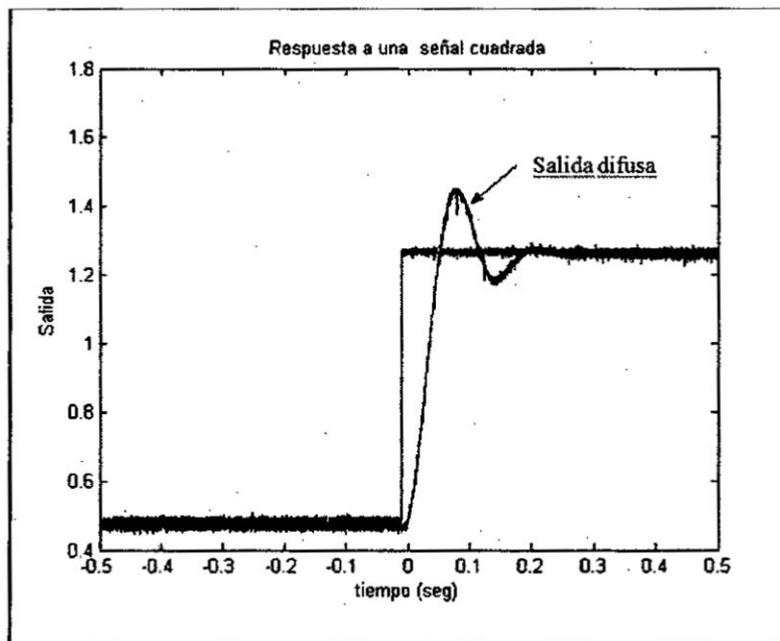
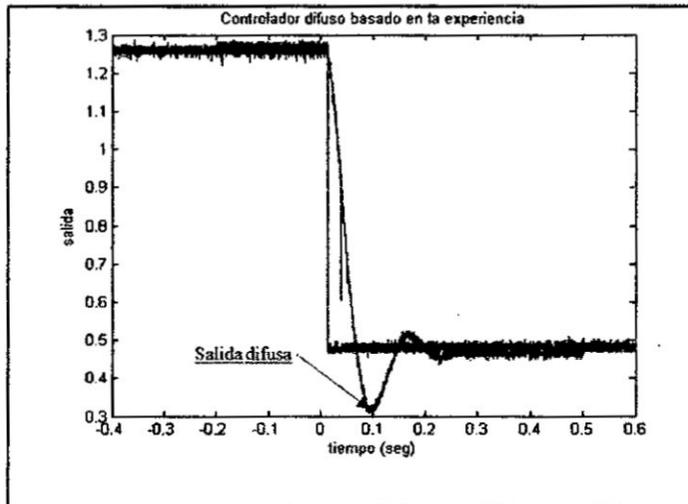


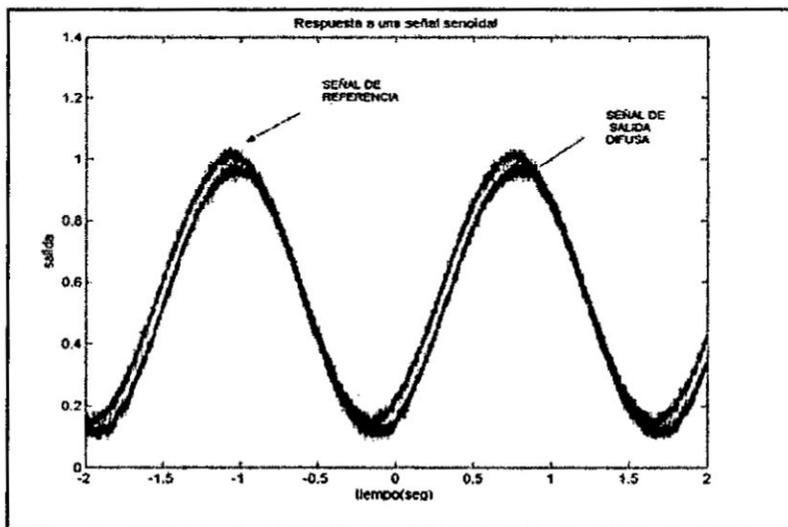
Figura 5.2.1 Respuesta temporal del servomecanismo a una señal de Referencia cuadrada, parte de subida

En La figura 5.2.2 se presenta la señal de salida del sistema de control de posición, que corresponde a la parte de bajada de la señal de referencia.



**Figura 5.2.2** Respuesta temporal del servomecanismo a una señal de Referencia cuadrada, parte de bajada

En la figura 9.3 se presenta la señal de salida del sistema de control de posición, que corresponde a la parte de bajada de la señal sinusoidal de referencia. En Esta figura se aprecia un ligero error en estado estable, cuando la señal de referencia pasa del valor máximo al valor mínimo.



**Figura 5.2.3** Respuesta temporal del servomecanismo a una señal de Referencia sinusoidal

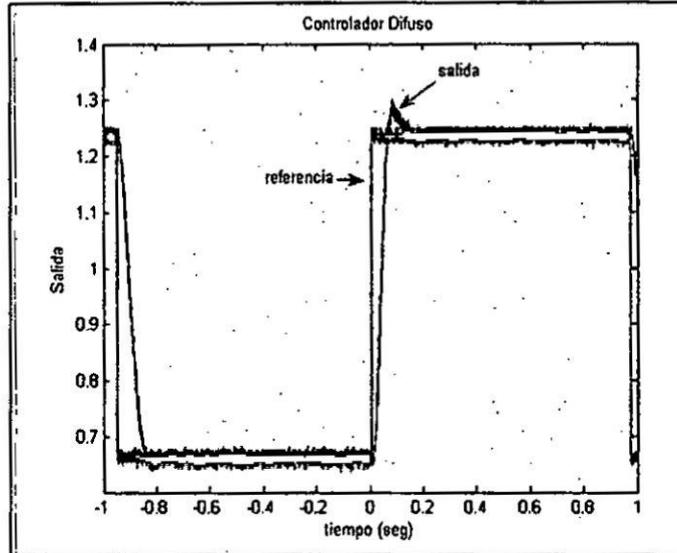
## RESULTADOS DEL CLD SINTONIZADO EN BASE AL MÉTODO DE OPTIMIZACIÓN

En la Tabla se muestra la base de conocimiento del controlador difuso que fue obtenida con el método de optimización; la base de reglas del controlador difuso, está formada por 49 reglas de inferencia; Las reglas de inferencia difusa se pueden obtener de la Tabla, de la misma manera como fue explicado para el caso del CLD sintonizado en base a la experiencia.

| <i>VELOCIDAD</i> → |    |    |    |    |    |    |    |
|--------------------|----|----|----|----|----|----|----|
| <i>POSICION</i> ↓  | PG | PM | PP | CE | NP | NM | NG |
| PG                 | PG | PM | PP | PG | PG | PP | PM |
| PM                 | PG | PG | PG | PP | PM | PM | PM |
| PP                 | PM | PM | PG | PM | PM | PM | PM |
| CE                 | NM | NM | NP | CE | PP | PP | PP |
| NP                 | NG | NM | NP | NM | NP | NP | NP |
| NM                 | NP | NM | NP | NM | NP | NP | NP |
| NG                 | NG | CE | PG | NG | NP | NM | NG |

**Tabla Matriz de reglas en base a optimización**

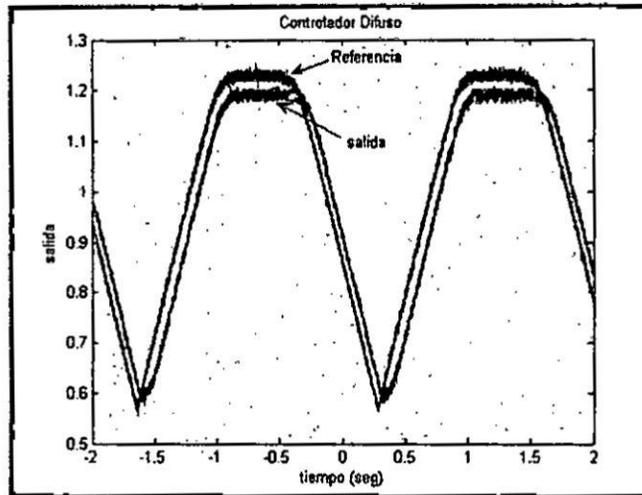
En la figura 5.4 se presenta la salida del servomecanismo cuando se aplica una señal cuadrada en la entrada de referencia. En esta figura se puede apreciar un sobrepaso pequeño en la parte de subida de la señal, aunque el tiempo de establecimiento es muy corto (aproximadamente 150 mseg.); En la parte de bajada de la señal de salida, se observa que no existe sobrepaso y el tiempo de establecimiento es de aproximadamente 150 mseg, lo cual nos recuerda que el controlador difuso es un sistema no lineal. Cabe aclarar que el pico que se aprecia en la señal de salida se debe al ruido generado por las conexiones temporales durante la prueba.



**Figura 5.4 Respuesta temporal del servomecanismo a una señal de Referencia cuadrada**

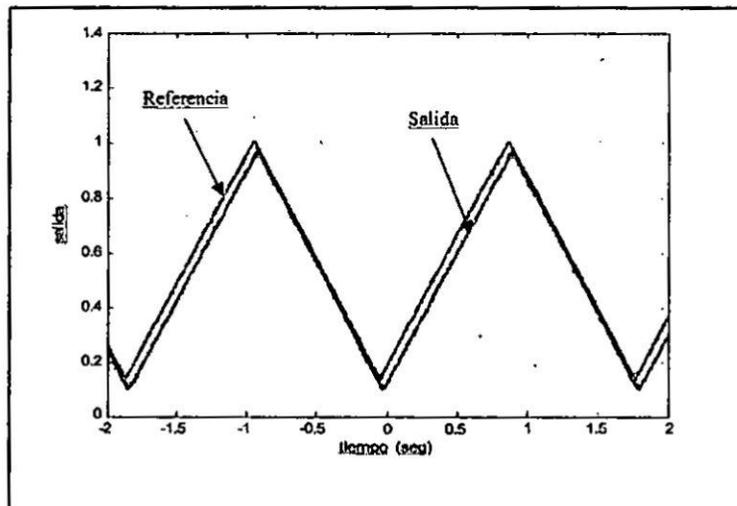
En la figura 5.4 se muestra un excelente desempeño del controlador difuso diseñado con el método de optimización, comparado con el CLD diseñado en base a la experiencia. Sin embargo este último puede ser mejorado sustancialmente, incrementado el número de reglas y ajustando las mismas en base a prueba y error.

En la figura 9.5 se muestra una señal triangular recortada, donde se aprecia error en estados estable.



**Figura 5.5 Respuesta temporal del servomecanismo a una señal de Referencia triangular recortada**

Finalmente, En la figura 5.6 se muestra una señal triangular, donde también se aprecia un ligero error en estados estable.



**Figura 5.6 Respuesta temporal del servomecanismo a una señal de Referencia triangular**

## **CAPITULO VI**

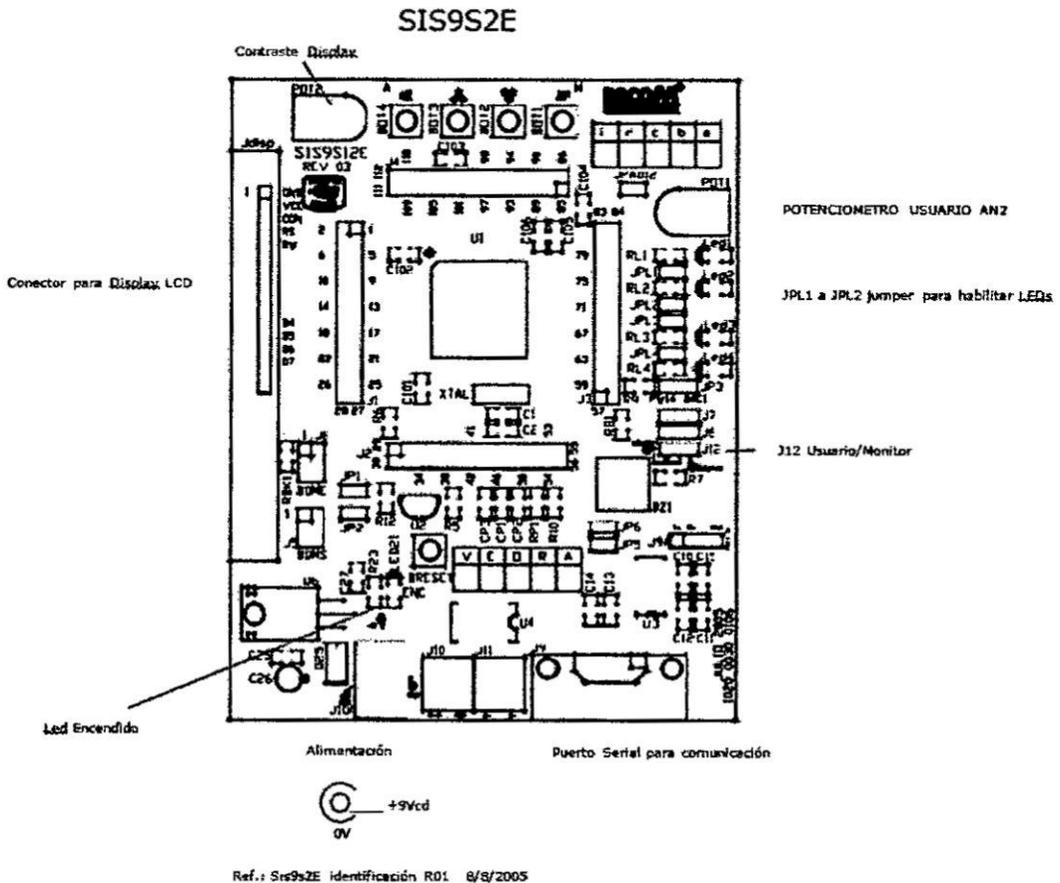
### **CONCLUSIONES**

En base a los resultados obtenidos, se puede concluir que el método de optimización, para obtener la base de reglas del controlador difuso es satisfactorio, teniendo en cuenta que se usó un algoritmo genético simple de aplicación general. De los resultados experimentales se puede observar que el controlador lógico difuso diseñado, presenta un buen desempeño, como se puede apreciar en la figura 5.4. La respuesta del sistema de lazo cerrado, a los cambios de referencia, son rápidos; sin embargo en la figura 5.4, se observa que se obtiene una mejor respuesta en la parte de bajada de la señal, ya que la salida sigue a la referencia de forma suave y rápida, comparada con la parte de subida de la señal de referencia, donde se presenta un sobrepaso pequeño, aunque la respuesta completa no presenta error en estado estable. La respuesta del sistema real tiene un mejor desempeño que el que se presenta en las simulaciones, esto es debido a que el ajuste fino de la base de reglas, del controlador difuso se realizó, ajustando las reglas difusas, durante el proceso de las pruebas experimentales, tomando como base las reglas obtenidas mediante el método de optimización. Como pruebas complementarias del CLD se aplicaron otras señales de prueba como son las señales triangular recortada y triangular, las cuales se pueden apreciar en las figuras 5.5 y 5.6 respectivamente. En las figuras 5.5 y 5.6 se observa un buen seguimiento de la señal de referencia, aunque se presenta un ligero error en estado estacionario. Lo anterior es debido a que el diseño del CLD se realizó para el seguimiento de señales cuadradas.

Una desventaja de este método para el diseño de controladores difusos es el tiempo de proceso, debido a que cada uno de los controladores que son evaluados, requiere de un tiempo de simulación.

En trabajos futuros se puede aplicar el procedimiento de optimización mostrado, a procesos más complejos que el presentado en esta investigación, donde se justifique plenamente la inversión de tiempo de máquina en el proceso de sintonización.

Para implantar físicamente el sistema de control, se requiere de un microcontrolador capaz de soportar instrucciones de lógica difusa, en este caso el microcontrolador MC9s12E128 con núcleo HCS12 de Motorola.



**Figura 1 Vista esquemática del microcontrolador mc9s12e**

Este microcontrolador posee cuatro instrucciones que efectúan los pasos necesarios en un controlador difuso (MEM, REV, REVW, WAV) las cuales determinan el grado de pertenencia, efectúan la evaluación de reglas y la defusificación. Además de éstas, existen más de 100 instrucciones que permiten al programador, implantar lazos de control en aplicaciones reales.

Cuando se habla de sistemas de control es prácticamente inherente hablar de señales externas o del mundo real. Es decir para efectuar hasta el más simple lazo de control es necesario obtener una señal del proceso a controlar. En este caso se

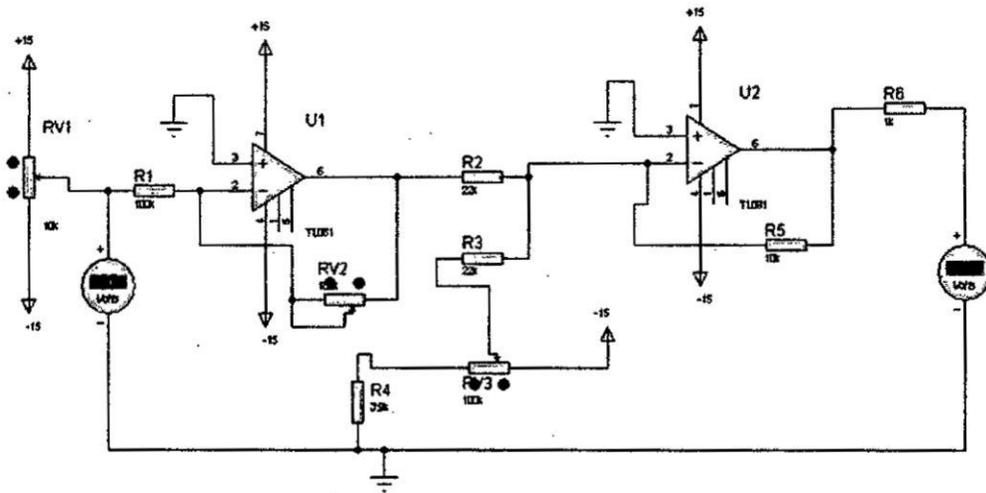
necesita traducir la posición del servomotor a una señal eléctrica proporcional al desplazamiento angular, para esto, la planta está equipada con un potenciómetro que proporciona un voltaje variable dependiendo de la posición del eje del motor. Para discretizar esta variable se usó el convertidor analógico-digital del propio microcontrolador.

Efectuando este proceso de conversión Analógico-digital, el controlador difuso obtiene las entradas *crisp*, es decir el sistema ya puede generar una respuesta a partir de condiciones reales. Pero esta respuesta, está aún dentro del microcontrolador, es decir, es aún, una representación binaria que poco significa para el proceso o la planta, es necesario entonces, de otro de los recursos del microcontrolador, el convertidor digital-analógico. Con este subsistema se pueden eliminar componentes externos y así, eliminar posibles errores en el sistema de control.

Siguiendo con las características del microcontrolador cabe destacar sus limitaciones, aquellas que se refieren a las tensiones apropiadas para su funcionamiento correcto. Debido a que trabaja a niveles de voltaje del tipo TTL, los valores analógicos que maneja solo pueden fluctuar alrededor de estos valores, es decir de 0 a 5 voltios aproximadamente. Es en este momento que se requiere de algún tipo de acondicionamiento de señal; esto se solucionó utilizando los amplificadores operacionales en sus diversas configuraciones (AO's). Utilizando los AO's también se soluciona cualquier efecto de carga indeseable que pudiera estar propiciando alguno de los sensores o el actuador mismo.

El CLD requiere de dos variables de entrada, dos de las características dinámicas del sistema, que en este caso son la posición y velocidad del motor. En el caso de la posición, el sensor es un potenciómetro de 5 K $\Omega$ . En el caso de la velocidad, el sensor es un tacómetro. Con esta señal se tuvo especial cuidado debido a los cambios de sentido y las aceleraciones crecientes debido a que el voltaje pasaba de un valor pequeño a uno grande y de signo contrario y esto dañaría el convertidor del microcontrolador. Para corregir esto se utilizó una etapa de acondicionamiento a modo de sumador inversor como se muestra en La figura de 2. El primer AO funciona como atenuador inversor de la tensión de entrada y el segundo es el sumador inversor. A la salida de esta configuración de amplificador obtenemos un voltaje variable de 0 a 5

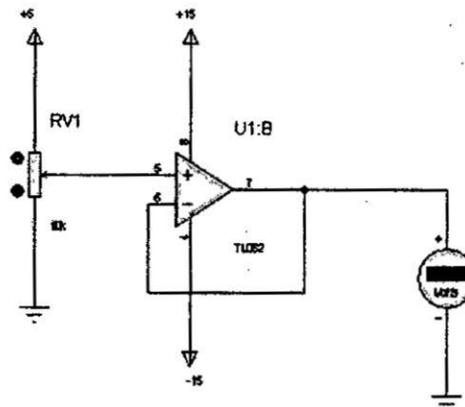
voltios.



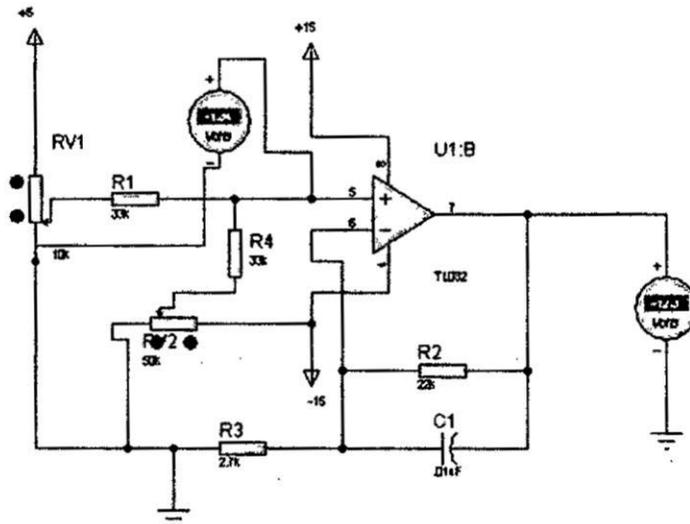
**Figura 2 Acondicionamiento de la señal de velocidad**

Como se mencionó anteriormente, el sistema produce una respuesta a las condiciones actuales del sistema. Esta respuesta o voltaje no es el adecuado en valores de tensión ni de potencia para alimentar al motor del sistema. Se necesita de un sistema capaz de llevar los 0 y 5 voltios máximos, a niveles de (+)VCC y (-)VCC, para que el módulo de potencia opere satisfactoriamente.

En las Figuras 3 y 4, se muestran estas configuraciones. El primer módulo funciona como seguidor de tensión, para acoplar las impedancias entre el convertidor digital-analógico del microcontrolador y la interfaz del segundo módulo, el cual es un sumador no inversor.



**Figura 3 Acondicionamiento de la señal de salida**



**Figura 4 Acondicionamiento de la señal de salida (sumador no inversor)**

La familia de microcontroladores MC9s12E es una familia de bajo costo de propósito general, la cual tienen periféricos estándar, incluyendo la unidad central de procesamiento (CPU) HCS12, tiene 128K de flash, 8K de RAM, tres puertos Seriales SCI, un SPI, una interfaz *buc* IIC(IIC), tres temporizadores (TIM) de cuatro canales cada uno, 6 canales PWM de 15 bits con módulo de protección por Falla (PMF), 16 canales del convertidor analógico digital, 16 ADC, y 2 convertidores de digital a analógico de 8 bits (DAC), Incluye un circuito PLL, que permite a cada periférico un ahorro de energía de desempeño, según la aplicación. En adición a los puertos I/O disponibles en cada módulo, cuenta con 16 puertos I/O con la característica de despertar (Wake-up), del modo de operación STOP o WAIT, el integrado incluye un regulador interno que permite su operación de 2.5V de un fuente externa de 3.15 a 5.5V.

La comunicación entre el sistema y la computadora se realiza a través del programa: *Serial monitor*, que está previamente cargado en el sistema de desarrollo que permite la programación y depuración de programas a través del puerto serie.

El programa base es el CodeWarrior™ (Integrated Development Environment (IDE)), que provee un sistema de desarrollo de software flexible y eficiente.

El código fuente está hecho en lenguaje ensamblador, como se muestra a continuación:

## CARACTERÍSTICAS DEL SERVOMOTOR EXPERIMENTAL MARCA DALVIN-1000

### Motor/ Carga

#### Parámetros del motor

|                                   |                               |
|-----------------------------------|-------------------------------|
| Resistencia de armadura           | 3.77 Ohms                     |
| Inercia del motor                 | 0.0054 Oz-in-seg <sup>2</sup> |
| Viscosidad del motor              | 0.5445 oz-in/rad/seg          |
| Constante de fuerza electromotriz | 0.12032 volts/rad/seg         |
| Constante del par                 | 0.12032 volts/rad/seg         |

#### Parámetros de la carga

|                      |                      |
|----------------------|----------------------|
| Inercia de la carga  | 17.0 oz-in/amp       |
| Factor de viscosidad | 0.0010 oz-in/rad/seg |

#### Sensores Velocidad

|                              |                              |
|------------------------------|------------------------------|
| Frecuencia de rizo           | 11.0 Hz/rev                  |
| Inercia de armadura          | 0.001 oz-in-seg <sup>2</sup> |
| Linealidad                   | 0.2%                         |
| Salida del tacómetro         | 0.003 volts/rpm              |
| Resistencia de Terminal      | 2200 ohms                    |
| Inductancia                  | 0.0026 henries               |
| Velocidad máxima             | 20000 rpm                    |
| Resistencia de carga nominal | 11500 Ohms                   |
| Posición                     |                              |
| Tipo                         | Plástico conductivo          |
| Resistencia                  | 5000 Ohms                    |

Rotación

Sin límite

Características físicas

Dimensiones

17"X 8" X 10"

Peso

12 lbs.

## CAPITULO VII

### REFERENCIAS

- [1] Ash R. *Real Analysis and Probability*, Ed. Academic Press, USA, 1970.
- [2] Åström, K.J. and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, Second Edition, Instrument Society of America, Research Triangle Park, NC, 1995.
- [3] Bäck Thomas, *Evolutionary Algorithms in theory and practice*, Oxford University Press, 1996
- [4] Chi-Tsong Chen, *Linear System Theory and Design*, Oxford University Press, Third Edition, 1999
- [5] Coello-Coello Carlos A., David A. Van Veldhuizen, and Gary B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, 2002.
- [6] Carlos A. Coello-Coello, "An Update Survey of Evolutionary-Based Multiobjective Optimization Techniques: State of the Art and Future Trends". In 1999 Congress on Evolutionary Computation. Washington, D.C. IEEE Service Center, 1999, pp. 3-13.
- [7] CPU12 Reference Manual, MOTOROLA Inc. 1997
- [8] *Fuzzy Logic Toolbox user's guide*, Mathworks Inc., USA, 2007.
- [9] Holland John H. *Adaptation in Natural and Artificial Systems*, Massachusetts Institute Technology, Third printing, 1994.
- [10] Hopgood Adrian A., *Intelligent Systems for Engineers and Scientists*, CRC press, 2001
- [11] Jing Yuan Zhang and YeDe Li, "Application of Genetic Algorithm in optimization of Fuzzy Control Rules", Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, (ISDA06), 2006.
- [12] Karr C.L., "Design of a Adaptive fuzzy logic controller using a genetic algorithm", Proceedings of the Fourth International Conference, Belew, R. and Brooker, L, Eds., Morgan Kaufmann, San Mateo California, 1991.
- [13] Kuri Morales Angel, *A Comprehensive Approach to Genetic Algorithms in Optimization and Learning Theory and applications*, Instituto Politecnico Nacional, México, volume 1, 1999

- [14] Lagunas R., Fernández\_Anaya G. and Martínez García J.C. "Experimental Evaluation of a Mixed  $H_2/H_\infty$ -Based PID Using Genetic Algorithms", Proceedings of the IASTED International Conference, Circuits, Signals and Systems, May 19-21, Cancun México, 2003
- [15] Lagunas-Jiménez R. and Pitalúa-Díaz N. "Tuning Fuzzy Control Rules via Genetic Algorithms", Electronics, Robotics, and Automotive Mechanics Conference, Cuernavaca, Morelos, México, september (CERMA), 2007.
- [16] Lee C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic en Controller-Part 1", Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 1, pp. 404-418. Mar/Apr 1990
- [17] Lee C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic en Controller-Part ", Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp. 419-435, Mar/Apr 1990
- [18] Li Y. F. and Lau C.C., "Development of Fuzzy Algorithms for Servo Systems", IEEE Countr. Syst. Mag., vol. 9 No. 3 pp. 65-72, 1989
- [19] Liu G.P. Yang J.B. and Whidborne J.F., *Multiobjective Optimization and Control*, Research Studies Press LTD, England, 2003
- [20] Mamdani, E. H. "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant" Proc. IEE Vol 121, N° 12, pp 1585-1588, 1974.
- [21] Mamdani, E. H. and assilian "A Fuzzy Logic Controller for a Dynamic Plant" Intl. J. Man Machine stud., pp. 1-13, 1975.
- [22] Man K.F., Tang K.F. and Kwong S., *Genetic Algorithms*, Springer-Verlag, Great Britain, 1999
- [23] Mo Jamshidi, Leandro dos Santos Coelho, Renato A. Krohling and Peter J. Fleming, *Robust Control Systems with Genetic Algorithms*, CRC Press, 2003.
- [24] Passino M. K., Yurkovich S., *Fuzzy Control*, Addison-Wesley, USA, 1998.
- [25] Witold Pedrycz, *Fuzzy control and Fuzzy systems*, John Wiley & Sons Inc. second edition, 1996
- [26] Von Altrock C., *Fuzzy Logic and NeuroFuzzy Applications Explained*, Prentice Hall PTR, United States of America, 1995.
- [27] Zadeh.. L.A., "Fuzzy Sets", Information and Control, Vol. 8, No. 3 pp. 338-353, June 1965.
- [28] Zadeh L.A., "Communication Fuzzy Algorithms", Information and Control, Vol. 12 pp. 94-102 USA, 1968.

## CAPITULO VIII

### VIII. MATRIZ DE CONSISTENCIA.

#### " ALGORITMO GENETICO APLICADO AL CONTROL DIFUSO DE POSICION DE UN SERVOMOTOR".

| PROBLEMA  | OBJETIVOS   | HIPÓTESIS  | VARIABLES  | TIPO DE INVESTIGACIÓN  |
|---|---|--|--|--|
| <p><b>Problema principal</b></p> <p>¿Qué consideraciones se deben tomar para el desarrollo de un modelamiento de una unidad de Algoritmo Genetico Aplicado al Control Difuso de Posicion de un Servomotor?</p> <p><b>Problemas secundarios</b></p> <p>a.- ¿De qué manera un modelamiento de una unidad de control optimiza el trabajo en los motores?</p> | <p><b>Objetivo General</b></p> <p>Controlar las variables del proceso del sistema algoritmo genético aplicado al control difuso de posición de un servomotor disponible mediante técnicas de control moderno y técnicas de control clásico, para observar</p> | <p><b>Hipótesis principal</b></p> <p>Un sistema Algoritmo Genetico Aplicado al Control Difuso de Posicion de un Servomotor está destinados a la realización de pruebas en forma completamente automática, incluyendo la adquisición de datos, monitoreo local y remoto, presentación de valores en pantalla, y migración</p> | <p><b>Variable Independiente:</b></p> <p>La variable independiente, también conocida como variable manipulada, se encuentra en el centro de cualquier diseño experimental cuantitativo. Para nuestro presente trabajo de investigación la variable independiente</p> | <p><b>1. Tipo de Investigación</b></p> <p>Aplicada</p> <p><b>2. Nivel de investigación</b></p> <ul style="list-style-type: none"> <li>○ Descriptivo</li> <li>○ Explicativo</li> </ul> <p><b>3. Diseño</b></p> <p>Por objetivos</p> |

|  |   |   |   |  |
|--|---|---|---|--|
| <p>b.- ¿En qué medida se comprueba la importancia del Desarrollo de un modelamiento Algoritmo Genetico Aplicado al Control Difuso de Posicion de un Servomotor?</p> <p>c.- ¿De qué manera se identifica las limitantes en el Desarrollo de Algoritmo Genetico Aplicado al Control Difuso de Posicion de un Servomotor?</p> | <p>el desempeño dinámico de cada uno de los algoritmos implementados.</p> <p><b>Objetivos secundarios</b></p> <p>a.- Plantear las etapas del proceso de sistema algoritmo genético aplicado al control difuso de posición de un servomotor con un modelo matemático que la caracterice</p> <p>b.- Diseñar algoritmos de control mediante técnicas modernas de</p> | <p>de resultados de ensayos en bases de datos.</p> <p><b>Hipótesis secundarias</b></p> <p>a. El desarrollo de un modelamiento de una unidad de control incide en el manejo más eficiente de un motor, optimizando mejor los recursos mediante los parámetros obtenidos.</p> <p>Mejorando el monitoreo y control de los motores de combustión.</p> <p>Produciendo cambios significativos a la hora del manejo.</p> | <p>es: <b>EI ALGORITMO GENETICO.</b></p> <p><b>Variable Dependiente:</b><br/>En cualquier experimento real, un investigador manipula una variable independiente para influir en una o varias variables dependientes. Para nuestro presente trabajo de investigación tenemos las siguientes variables dependientes que son el uso de muestras de recolección de datos.</p> | <p><b>4. Población</b></p> <p>N = 10 000</p> <p><b>5. Muestra</b></p> <p>La muestra es:</p> <p>n = 49</p> <p><b>6. Técnicas de recolección de datos</b></p> <ul style="list-style-type: none"> <li>○ Encuestas</li> <li>○ Análisis de datos.</li> <li>○ Análisis documental.</li> <li>○ Revisión documental</li> </ul> <p><b>7. Instrumentos de recolección de datos</b></p> |
|--|---|---|---|--|

|  |  |   |  |  |
|--|--|---|--|--|
|  | <p>control como lógica difusa y optimización por medio de algoritmos genéticos y técnicas clásicas de control como (PID) para la etapa seleccionada del proceso</p> <p>Definir las métricas (por ejemplo error estado estacionario) para la comparación del desempeño dinámico de los controladores, y aplicarlas para cada uno de los controladores</p> | <p>b. La importancia del desarrollo del desarrollo de un Algoritmo Genético Aplicado al Control Difuso de Posición de un Servomotor estimula a los estudiantes a buscar mejores desarrollos y aplicaciones para dicho modelo.</p> |  | <ul style="list-style-type: none"> <li>○ Ficha bibliográfica</li> <li>○ Guía de entrevista.</li> <li>○ Ficha de encuesta.</li> </ul> |
|--|--|---|--|--|