

Implementación de Hardware / Software para Control Difuso

*M. en C. Romeo Urbieto Parrazales,
M. en C. Pablo Manrique Ramírez,
Ing. Antonio Hernández Zavala;
Profesores del CIC-IPN
e-mail: rurbieta@cic.ipn.mx*

Este artículo trata el diseño e implementación de hardware/software de control difuso usando tecnología avanzada de microcontroladores. El diseño de control difuso usa un esquema de dos variables de entrada: el error de velocidad y su respectiva derivada, y una salida, el voltaje de armadura del motor. Las variables de entrada fueron convertidos a valores difusos en tres funciones de membresía triangulares cada una, mientras la variable de salida en cinco funciones de membresía puntuales. La base del conocimiento viene expresada mediante el método de la composición max-min, produciendo una matriz de 4x4 reglas. La implementación de hardware viene representada en un diagrama a bloques: un microcontrolador cop8cdr como núcleo del sistema de hardware y sus respectivos periféricos: teclado, despliegue y adquisición, mientras que el software está representado en diagramas de flujo de las rutinas de control y periféricos.

INTRODUCCIÓN

Un sistema es un conjunto de procesos, desde los más simples hasta los más complejos, con entradas, funciones y salidas.^[1]

Con el paso del tiempo surgen diversas teorías que se encargan de mejorar el desempeño de los sistemas; en el caso de sistemas inciertos (ya sean industriales, humanos o naturales) existe la Lógica Difusa, que es una herramienta matemática para modelación o control de los procesos.^[2]

Aunque en este artículo se utilizan la Lógica y el Control Difuso como herramientas para control de procesos eléctricos; no se abordará la base teórica en la que

se sustentan. Sin embargo, se detalla la realización de un Sistema de Control Difuso que maneja un motor de corriente directa implementado en un microcontrolador.^[3]

Todo proceso de control, requiere que el usuario asigne un valor de operación para la variable a controlar, debe existir también un sistema que lleve al proceso a las condiciones indicadas por el usuario de modo fácil y preciso .

Para lograr lo anterior, se utiliza un motor de CD y un tacómetro analógico; en el cual mediante un dispositivo electrónico, el usuario define la velocidad a la que desea opere el motor. Posteriormente, un dispositivo se encarga de manejar el motor de forma automática según lo especificado manteniendo la velocidad hasta que el usuario determine lo contrario.^[4]

DISEÑO

Problema. Se desea controlar la velocidad de un motor de corriente directa, mediante una interfaz de usuario electrónica; para conocer la velocidad actual del motor se cuenta con un tacómetro analógico que mediante un transductor entrega la lectura. Para asignar el voltaje de operación en la entrada del motor se cuenta con un transductor de entrada.

Análisis. Como se puede observar, mediante una señal de entrada en el rango de 0 a 6 Vcc, se puede controlar al motor a velocidades en el rango de 0 a 2000@Rpm. Debido a las especificaciones del motor no se deben rebasar las 1700@Rpm. Por ello, el motor será operado en el rango de 0 a 5 Vcc. El valor equivalente en voltaje a la velocidad actual del motor, es entregado por el driver del tacómetro en el que la salida se encuentra en el mismo rango de voltaje de entrada, por lo que para la entrada y la salida se utilizarán los valores de la **Tabla 1**.^[5]

Vin REF	Motor (@Rpm)	Vout TACH	Valor decimal
0	0	0	0
0.25	0	0.25	12
0.5	0	0.5	25
0.75	36	0.75	38
1	160	1	51
1.25	260	1.25	63
1.5	360	1.5	76
1.75	460	1.75	89
2	560	2	102
2.25	660	2.25	114
2.5	760	2.5	127
2.75	860	2.75	140
3	960	3	153
3.25	1060	3.25	165
3.5	1160	3.5	178
3.75	1260	3.75	191
4	1360	4	204
4.25	1460	4.25	216
4.5	1560	4.5	229
4.75	1660	4.75	242
5	1760	5	255

Tabla 1. Valores de entrada y salida del proceso.

En la tabla, la señal de entrada para el sistema analógico del motor recibe el nombre de OUT, al dato de salida del tacómetro se le asignará el nombre de TACH, y al dato de referencia asignado por el usuario se le llamará REF. Estos son los nombres de variable que utilizará el controlador.

En el diagrama general (**Figura 1**) se puede notar que el modelo de control es de lazo cerrado, esto es, el dato de salida del controlador depende directamente del dato de

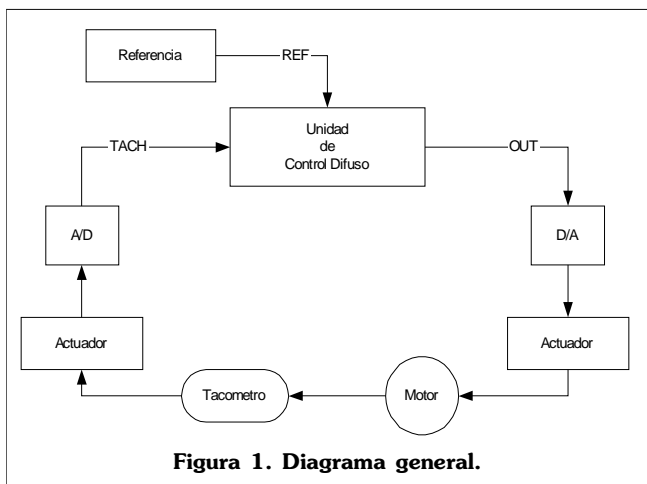


Figura 1. Diagrama general.

entrada; a partir de esta referencia y el dato tomado del tacómetro por el convertidor A/D (TACH) se obtiene un error, el cual se usa en el algoritmo de control difuso para obtener una salida de control que será enviada a través de un convertidor D/A hacia el motor, repitiendo iterativamente el procedimiento anterior.

Cabe señalar que solo será tratada la parte digital, misma que comprende los conversores A/D y D/A, la unidad de control difuso y la referencia de usuario, por lo que el motor, tacómetro y actuadores no se abordarán a detalle; esto es, se presupone su conocimiento.^[6]

El Sistema recibe un par de datos de entrada provenientes del exterior; el primero de ellos es el introducido por el usuario REF y el segundo es la velocidad actual del motor dada por el tacómetro TACH.

Al restar al valor de referencia el dato del tacómetro, se obtiene el valor del ERROR,

$$ERROR = REF - TACH$$

La derivada del error se obtiene de la siguiente operación

$$DERROR = ERROR_i - ERROR_{i-1}$$

Donde $ERROR_{i-1}$ es el error en el instante anterior, que se asigna a la variable AUX; el cálculo de la derivada del error permite al controlador seguir la ruta correcta de evaluación. Lo anterior se puede apreciar en la **Figura 2**.

Control Difuso. El Control Difuso, esta compuesto por tres etapas: la primera es la fuzzificación, aquí se convierten los valores de entrada a valores difusos; la segunda es la evaluación de los valores fuzzificados en la base de reglas de control; y la última es la desfuzzificación, en la que se obtiene un valor de salida no difuso. Una vez identificado el modelo del control difuso, se puede detallar ampliamente el sistema a realizar.^[7]

La **Figura 3**, muestra las partes que componen al sistema en su totalidad.

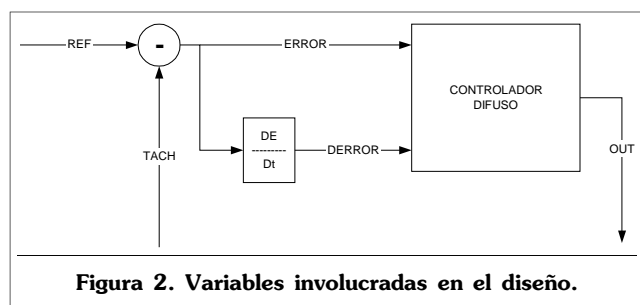


Figura 2. Variables involucradas en el diseño.

Fuzzificación. Un controlador difuso utiliza números difusos de la forma $A_\alpha = (x, \mu_\alpha(x))$, para los que a cada valor de entrada x , le corresponde un valor de $\mu_\alpha(x)$. Para las variables de entrada ERROR y DERROR, se tienen los siguientes números difusos con sus correspondientes variables lingüísticas (ver también las Figuras 4 y 5).

Para ERROR:

ERROR-NEGATIVO

$$EN = 1 - \left(\frac{x + 255}{255} \right), [-255, 0] \quad \dots (E0)$$

ERROR-ZERO

$$EZ = \frac{x + 55}{55}, [-55, 0] \quad \dots (E1)$$

$$EZ = 1 - \frac{x}{55}, [0, 55] \quad \dots (E2)$$

ERROR-POSITIVO

$$EP = \frac{x}{255}, [0, 255] \quad \dots (E3)$$

Para DERROR:

DERIVADA_ERROR-NEGATIVA

$$DEN = 1 - \left(\frac{x + 255}{200} \right), [-255, -55] \quad \dots (DE0)$$

DERIVADA_ERROR-ZERO

$$DEZ = \frac{x + 110}{110}, [-110, 0] \quad \dots (DE1)$$

$$DEZ = \frac{x - 110}{-110}, [0, 110] \quad \dots (DE2)$$

DERIVADA_ERROR-POSITIVA

$$DEP = \frac{x - 55}{200}, [55, 255] \quad \dots (DE3)$$

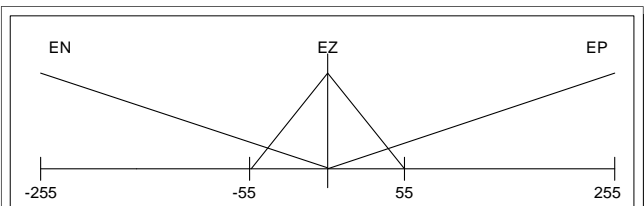


Figura 4. Variable lingüística del error.

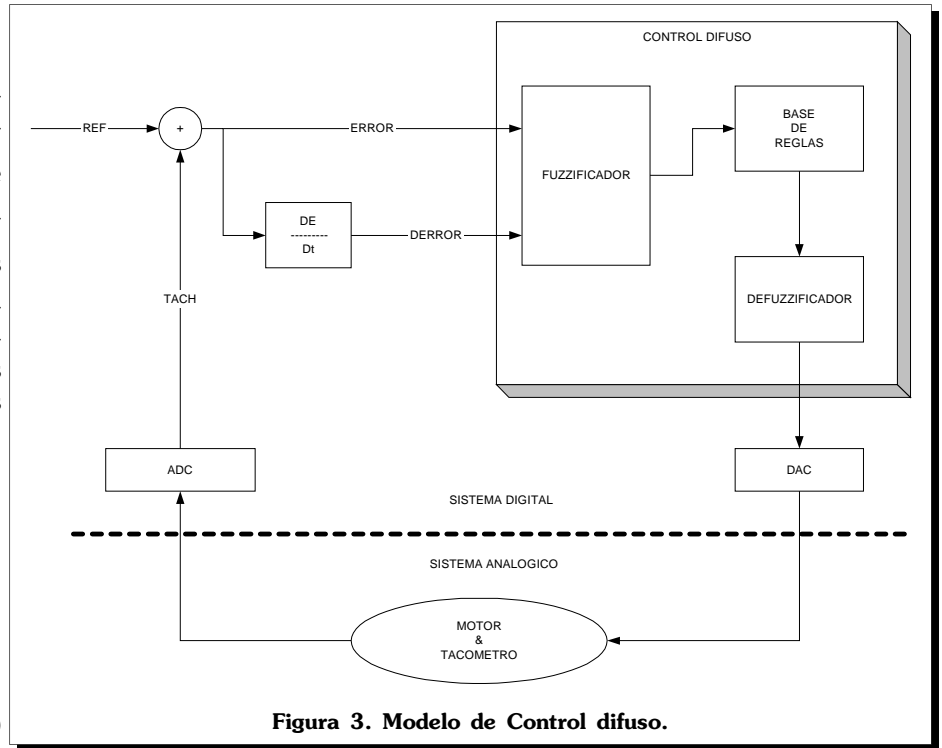


Figura 3. Modelo de Control difuso.

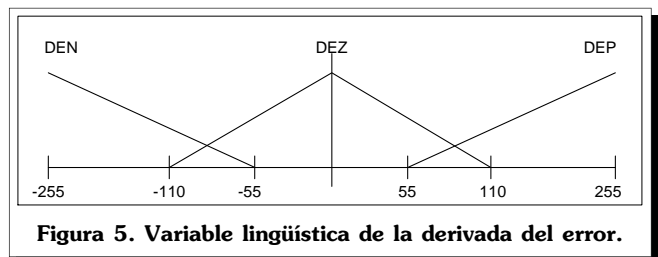


Figura 5. Variable lingüística de la derivada del error.

Se utilizan un par de vectores que contienen los datos calculados en esta etapa, con el propósito de poder realizar el método de composición; cada localidad corresponde a la fórmula que se indica. Los vectores se muestran a continuación.

ERROR

Posición	0	1	2	3
Formula	EO	E1	E2	E3

DERROR

Posición	0	1	2	3
Formula	DE0	DE1	DE2	DE3

Método de Composición max-min. Para evaluar un dato en la base de reglas, se debe contar con un método que permita elegir de entre el conjunto de datos de entrada, cuál será utilizado; para esto se realiza la composición MAX-MIN entre ERROR y DERROR, el dato obtenido debe indicar qué posición de la base de reglas se debe utilizar.^[8]

Esta composición, mediante un proceso similar a la multiplicación de matrices, produce como resultado una matriz de 4x4 (Tabla 2), mediante evaluación de mínimos y máximos de acuerdo a la siguiente formula:

$$E \circ DE = \left\{ (x, z), \max_y (\min(\mu_E(x, y), \mu_{DE}(y, z))) \right\}$$

	0	1	2	3
0	AL00	AL01	AL02	AL03
1	AL10	AL11	AL12	AL13
2	AL20	AL21	AL22	AL23
3	AL30	AL31	AL32	AL33

Tabla 2. Matriz alfa resultante.

Dado que la composición dejará un solo resultado diferente de cero, este corresponde a la localidad en la base de reglas que determinará la salida. La base de reglas utilizada se muestra en la Tabla 3.

	0	1	2	3
0	NM	NM	NM	NL
1	PM	Z	Z	NM
2	PM	Z	Z	NM
3	PL	PM	PM	PM

Tabla 3. Base de reglas óptimas.

A cada una de las variables, le corresponde un valor de salida según se muestra:

- NL = -64
- NM = -31
- Z = 2
- PM = 31
- PL = 64

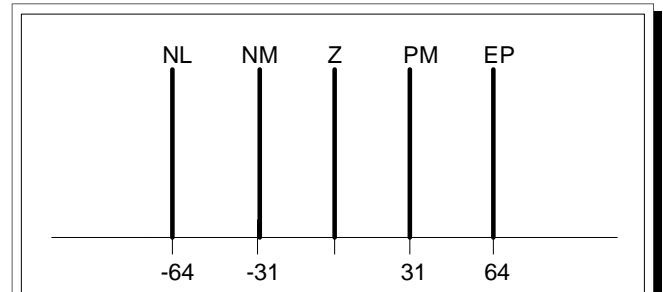


Figura 6. Variable lingüística de salida.

El resultado se carga en la variable de programa RES. La Figura 6 es la forma gráfica de representación de los resultados de salida.

Desfuzzificación. Esta etapa se encarga de sumar o restar, según el caso, el valor obtenido de la etapa anterior con el valor de la salida actual.

$$OUT = \sum RES$$

IMPLEMENTACIÓN DE HARDWARE/SOFTWARE

El sistema se implementará en un microcontrolador COP8 con convertor A/D integrado, la salida será dada a través de un convertor D/A y los resultados se presentarán en una pantalla LCD.

El diagrama general del sistema controlador difuso en hardware se muestra en la Figura 7.

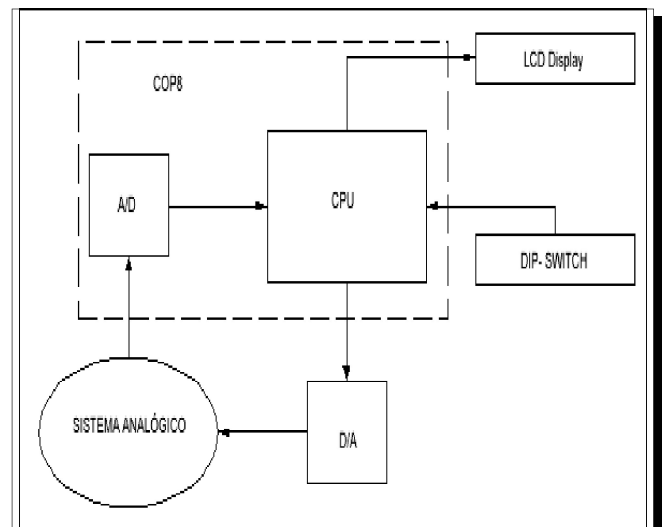


Figura 7. Hardware del sistema de control difuso.

De manera general, el algoritmo que se utiliza es el siguiente:

1. Lee referencia del usuario.
2. Lee dato del ADC.
3. Cálculo de error y derivada del error.
4. Fuzzificación.
5. Composición MAX – MIN.
6. Evaluación de reglas.
7. Defuzzificación.
8. Envío de salida a DAC.
9. Repite desde 1.

En la **Figura 8** se muestra el diagrama de flujo del programa y en la **Figura 9** se tiene el diagrama de conexiones del sistema.

• FUNCIÓN DE LECTURA DE REFERENCIA.

Para esta función se requiere leer un dato que el usuario ha colocado en el puerto de entrada, al momento de presionar un botón. Para la entrada se utilizan los bits de acuerdo a la tabla siguiente:

Puerto	A1	A2	A3	A4	A5	A6	A7	E0
REF	0	1	2	3	4	5	6	7

Para leer el dato se requiere realizar un desplazamiento en A, y probar el bit 0 de E.

Un botón conectado en el puerto G0, sirve como señal de interrupción; al generarse esta, el dispositivo realiza la rutina de servicio a interrupción externa, para obtener la lectura correspondiente.

• FUNCIÓN DE LECTURA DE DATO DE ADC.

El registro de configuración del convertidor integrado, cuenta con el bit ADBSY; el cual al ser establecido inicia la conversión, esto tarda un período de 15 ciclos de reloj, por lo que el algoritmo es el siguiente:

1. Establecer ADBSY.
2. Mientras ADBSY=1, esperar
3. Hacer TACH=ADRSTH
4. Fin.

• FUNCIÓN DE CÁLCULO DE ERROR Y DERIVADA DE ERROR.

El error es igual al dato de referencia menos la lectura del ADC; y la derivada del error es igual al error calculado menos el error anterior. Como el resultado puede ser positivo o negativo, existe un vector de signo (SIG-

NO), en donde el bit 7 corresponde al signo del error y el bit 6 al signo de la derivada de error.

• FUNCIÓN DE FUZZIFICACIÓN.

Dado que esta función será implementada en hardware las operaciones deben ser menos complejas, por lo que se utilizan las siguientes formulas equivalentes a las vistas anteriormente.

Puerto	D4	D5	D6	D7	E4	E5	E6	E7	F0	F1
DAC	D0	D1	D2	D3	D4	D5	D6	D7	WR	CS

E0
255-ERROR=RES
255-RES=RES

E1
55-ERROR=RES
RES x 4 = RES

E2
ERROR x 4 = RES
255 - RES = RES

E3
ERROR=RES

DE0

255 - DERROR = RES
255 - RES = RES

DE1
110 - DERROR = RES
RES x 2 = RES

DE2
DERROR x 2 = RES
255 - RES = RES

DE3
DERROR - 55 = RES

• FUNCIÓN DE COMPOSICIÓN MAX – MIN.

La composición entre los vectores ERROR y DERROR, utiliza apuntadores (EPTR, DEPTR respectivamente) a las localidades donde se encuentran los arreglos, y la matriz de alfa es apuntada por ALPTR. El método realiza las comparaciones entre ERROR X DERROR, colocando el resultado en la matriz de alfa; al final se realiza una segunda vuelta de búsqueda de máximo, para así solo tener un valor diferente de 0, que será el utilizado por la función siguiente.

• FUNCIÓN DE EVALUACIÓN DE REGLAS.

Esta función lee cada localidad de la matriz de alfa, si el contenido de ésta es diferente de cero, realiza la evaluación correspondiente, cargando en la variable RES el resultado obtenido; se utilizan la tabla y los valores de salida descritos anteriormente.

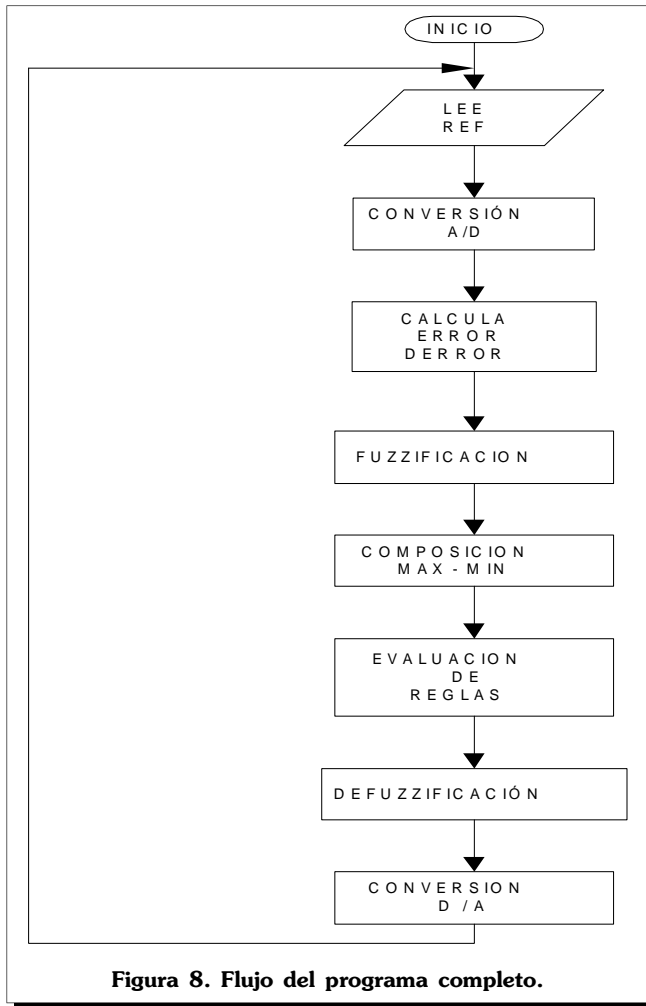


Figura 8. Flujo del programa completo.

• **FUNCIÓN DE DESFUZZIFICACIÓN.**

Mediante la utilización de RES y el bit de signo de ERROR, se realiza ya sea una suma o una resta de OUT +/- RES, utilizando OUT como un acumulador.

• **ENVÍO DE SALIDA A DAC.**

El contenido de OUT, se transfiere a puertos, la entrada del DAC, posteriormente se genera la señal de carga y retención de datos, con lo que se genera la señal de control en el voltaje deseado. Los puertos y las entradas al DAC se muestran en la tabla. Se utiliza un DAC TLC7524 configurado en modo voltaje.

CONCLUSIONES

1. El control difuso puede ser aplicado a cualquier problema que requiera la utilización de control.
2. El método MAX - MIN proporciona de manera fácil y precisa el resultado deseado de la base de reglas, pero si éste no está bien definido, conduce a errores.

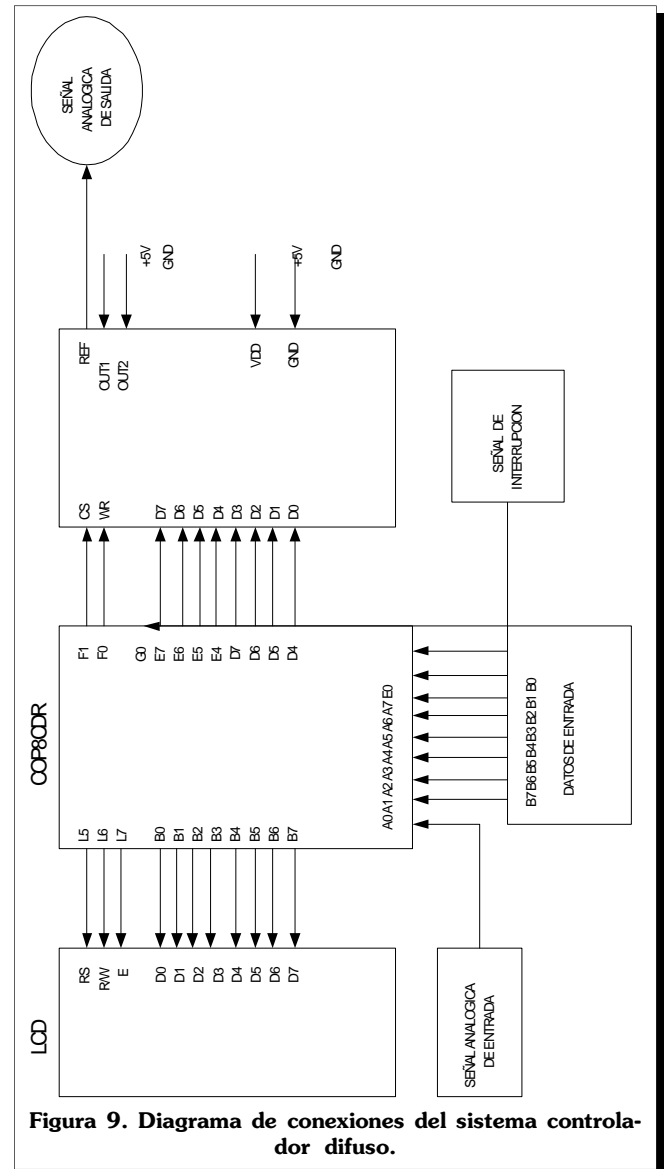


Figura 9. Diagrama de conexiones del sistema controlador difuso.

3. La definición de los números difusos de entrada es labor compleja, por lo cual se hace uso de otras herramientas matemáticas, tales como Redes Neuronales, Algoritmos Genéticos, etc.
4. No es necesario el uso de punto flotante como tal en las operaciones, se puede dar un resultado preciso trabajando con números enteros.
5. El verdadero poder de aplicación del control difuso se encuentra en problemas para los cuales las salidas no requieran un gran número de valores de salida, sino para los cuales exista un rango al cual se le pueda asignar un valor fijo.

REFERENCIAS

- [1] Peter Harriott. *Process Control. Capítulo 3. Open-Loop Response of Simple Systems*. Mc Graw-Hill Book Company. 1964.
- [2] George Bojadziev & Maria Bojadziev. *Fuzzy Sets, Fuzzy Logic, Applications. Capítulo 11. Fuzzy Logic and Applications*. World Scientific. 1995.
- [3] Jeremy Binfet & Bogdan M. Wilamowsky. *Micro-processor of Fuzzy Systems and Neural Network*. Página 234-239. 2001 IEEE.
- [4] José Ramón Alique López, Elena Agüero Gutiérrez, Leonardo C. Rosa. *A Fuzzy Logic Based Approach to Machine-Tool Control Optimization*. Páginas 524-539 del libro *Fuzzy Control Systems*. 1994.
- [5] Steve G. Romanink and Lawrence O. Hill. *Learning Fuzzy Control Rules from Examples*. Páginas 376-395 del libro *Fuzzy Control Systems*. 1994.
- [6] Ramiro Aduviri Velasco. *Comparación de Controladores Convencionales y Controladores Difusos*. Universidad de San Andrés. La Paz Bolivia. Agosto 2003.
- [7] George Bojadziev & Maria Bojadziev. *Fuzzy Sets, Fuzzy Logic, Applications. Capítulo 6 Fuzzy Sets*. World Scientific. 1995.
- [8] George Bojadziev & Maria Bojadziev. *Fuzzy Logic, Applications. Capítulo 7 Fuzzy Relations. Fuzzy Sets*. World Scientific. 1995.